

# Diskrétní Matematika (456-533 DIM)

Doc. RNDr. Petr Hliněný, Ph.D.

[petr.hlineny@vsb.cz](mailto:petr.hlineny@vsb.cz)

5. května 2006



Verze 1.02.

Copyright © 2004-2006 Petr Hliněný.

# Obsah

0.1	Předmluva . . . . .	iv
<b>I</b>	<b>Základy Diskrétní Matematiky</b>	<b>1</b>
<b>1</b>	<b>Množiny a výběry prvků</b>	<b>1</b>
1.1	Čísla, operace a množiny . . . . .	1
1.2	Výběry a seřazení prvků . . . . .	3
1.3	Výběry s opakováním . . . . .	6
1.4	Cvičení: Kombinatorické výběry . . . . .	8
<b>2</b>	<b>Diskrétní pravděpodobnost</b>	<b>9</b>
2.1	Motivační příklady . . . . .	10
2.2	Konečný pravděpodobnostní prostor . . . . .	10
2.3	Nezávislost jevů . . . . .	12
2.4	Střední hodnota . . . . .	13
2.5	Náhodné výběry . . . . .	15
2.6	Cvičení: Úlohy na pravděpodobnost . . . . .	16
<b>3</b>	<b>Důkazy v diskrétní matematice</b>	<b>18</b>
3.1	Základní logické symboly . . . . .	19
3.2	Pojem matematického důkazu . . . . .	19
3.3	Princip matematické indukce . . . . .	20
3.4	Důkazy vzorců pro výběry . . . . .	21
3.5	Vztahy s kombinačními čísly . . . . .	22
3.6	Důkazy “počítáním” . . . . .	24
3.7	Cvičení: Příklady matematických důkazů . . . . .	25
<b>4</b>	<b>Relace a zobrazení</b>	<b>27</b>
4.1	Pojem relace . . . . .	27
4.2	Uspořádání a ekvivalence . . . . .	28
4.3	Funkce, či zobrazení . . . . .	30
4.4	Skládání zobrazení; Permutace . . . . .	31
4.5	Princip inkluze a exkluze . . . . .	33
4.6	Cvičení: Příklady na relace a jiné . . . . .	34
<b>5</b>	<b>Algoritmizace diskrétních struktur</b>	<b>36</b>
5.1	Programová implementace struktur . . . . .	36
5.2	Implementace množin . . . . .	37
5.3	Generování výběrů . . . . .	39
5.4	Generátory náhodných čísel . . . . .	40
5.5	Kombinatorická “exploze” . . . . .	41
<b>II</b>	<b>Úvod do Teorie Grafů</b>	<b>42</b>
<b>6</b>	<b>Pojem grafu</b>	<b>42</b>
6.1	Definice grafu . . . . .	42
6.2	Stupně vrcholů v grafu . . . . .	44
6.3	Podgrafy a Isomorfismus . . . . .	45

6.4	Orientované grafy a multigrafy . . . . .	48
6.5	Implementace grafů . . . . .	49
6.6	Cvičení: Příklady o grafech a isomorfismu . . . . .	49
<b>7</b>	<b>Souvislost grafu</b>	<b>52</b>
7.1	Spojení vrcholů, komponenty . . . . .	53
7.2	Prohledávání grafu . . . . .	54
7.3	Vyšší stupně souvislosti . . . . .	55
7.4	Jedním tahem: Eulerovské grafy . . . . .	56
7.5	Cvičení: Příklady na souvislost grafů . . . . .	57
<b>8</b>	<b>Vzdálenost a metrika</b>	<b>60</b>
8.1	Vzdálenost v grafu . . . . .	61
8.2	Výpočet metriky . . . . .	62
8.3	Vážená (ohodnocená) vzdálenost . . . . .	63
8.4	Hledání nejkratší cesty . . . . .	64
8.5	Cvičení: Příklady o grafových vzdálenostech . . . . .	65
<b>9</b>	<b>Stromy a les</b>	<b>68</b>
9.1	Základní vlastnosti stromů . . . . .	69
9.2	Kořenové stromy . . . . .	71
9.3	Isomorfismus stromů . . . . .	73
9.4	Kostry grafů . . . . .	75
9.5	Cvičení: Příklady o stromech, kostrách a hladovém postupu . . . . .	78
<b>10</b>	<b>Barevnost a kreslení grafů</b>	<b>83</b>
10.1	Barevnost grafu . . . . .	83
10.2	Rovinné kreslení grafu . . . . .	85
10.3	Rozpoznání rovinných grafů . . . . .	88
10.4	Barvení map a rovinných grafů . . . . .	90
10.5	Cvičení: Příklady na barevnost a rovinnost grafů . . . . .	90
<b>11</b>	<b>Toky v sítích</b>	<b>93</b>
11.1	Definice sítě . . . . .	93
11.2	Hledání maximálního toku . . . . .	95
11.3	Zobecnění sítě a další aplikace . . . . .	97
11.4	Cvičení: Příklady toků v sítích . . . . .	100
<b>III</b>		<b>103</b>
	<b>Klíč k řešení úloh</b>	<b>103</b>
	Literatura . . . . .	109

## 0.1 Předmluva

Vážení čtenáři,

dostává se vám do rukou výukový text Diskrétní Matematiky, který je primárně určený pro studenty inženýrských oborů na technických vysokých školách.

Diskrétní matematika je moderní a rychle se rozvíjející oblastí matematiky, která má mnohé úzké vazby na teoretickou i aplikovanou informatiku. Kořeny diskrétní matematiky (kombinatoriky) sahají k velkým matematikům 18. a 19. století, z nichž si zaslouží jmenovat především Leonard Euler. Byl to však až nástup počítačů a rozvoj informatiky jako vědní disciplíny v druhé polovině 20. století, které přinesly nové teoretické pojmy a otázky a vtiskly tak diskrétní matematice její moderní tvář, spojenou dosti specificky s pojmem grafů.

Náš text vás seznámí jak s přehlednými základy klasické kombinatoriky, tak i se hlavními pojmy a mnoha prakticky aplikovatelnými poznatky teorie grafů. Je koncipován jako soběstačný celek, který od čtenáře nevyžaduje o mnoho více než běžné středoškolské matematické znalosti a chuť do studia. (Části věnované algoritmickým aplikacím navíc předpokládají znalost a zručnost v programování.) Svým rozsahem odpovídá jednomu semestru běžné výuky včetně cvičení.

Tento výukový text vychází z vynikající moderní učebnice [5] Kapitoly z Diskrétní Matematiky autorů J. Matouška a J. Nešetřila z MFF UK. (Tato kniha se kromě českého vydání dočkala i úspěšných vydání ve světových jazycích, jako například [6]. Vřele ji doporučujeme zájemcům o rozšíření svých matematických obzorů.) Záběr našeho textu není zdaleka tak široký jako u zmíněné učebnice, ale zaměřujeme se především na ty oblasti a úseky, které nacházejí nejčastější použití v inženýrské praxi. Na druhou stranu se zde více věnujeme přímým algoritmickým aplikacím uváděné látky a zmiňujeme také některé důležité detaily programátorské implementace.

Ve stručnosti se zde zmíníme o struktuře našeho textu. Přednesený materiál je dělený do jednotlivých lekcí (kapitol), které zhruba odpovídají obsahu týdenních přednášek v semestru. Lekce jsou dále děleny tematicky na oddíly. Výklad je strukturován obvyklým matematickým stylem na definice, tvrzení, případné důkazy, poznámky a neformální komentáře. Navíc je proložen řadou vzorově řešených příkladů navazujících na vykládanou látku a doplněn dalšími otázkami a úlohami. (Otázky a úlohy označené hvězdičkou patří k obtížnějším a nepředpokládáme, že by na ně všichni studující dokázali správně odpovědět bez pomoci.) Každá lekce je zakončena zvláštním oddílem určeným k dodatečnému procvičení látky. Správné odpovědi k otázkám a úlohám jsou shrnuty na konci textu.

Přejeme vám mnoho úspěchů při studiu a budeme potěšeni, pokud se vám náš výukový text bude líbit. Jelikož nikdo nejsme neomylní, i v této publikaci zajisté jsou nejasnosti či chyby, a proto se za ně předem omlouváme. Pokud chyby objevíte, dejte nám prosím vědět e-mailem <mailto:petr.hlineny@vsb.cz>.

*Autor*

## **Pár slov o vzniku**

Tento učební text vznikl v průběhu let 2003 až 2004 na základě autorových přednášek a cvičení předmětu Diskrétní matematika na FEI VŠB – TUO. Jeho základní verzi utvořil soubor autorových slidů pro přednášky předmětu. Většina řešených příkladů a úloh k řešení pak vychází ze skutečných písemných zápočtů a zkoušek v předmětu.

Za přípravu některých příkladů a zvláště za kontrolu celého textu patří dík také cvičícím – doktorandským studentům Martinu Kotovi, Ondřeji Kohutovi, Michalu Kolovratovi a Pavlu Moravcovi.

# Část I

## Základy Diskrétní Matematiky

### 1 Množiny a výběry prvků

#### Úvod

Základními stavebními kameny v matematice vůbec jsou čísla a množiny, přičemž v diskrétní matematice se převážně zabýváme přirozenými čísly a konečnými množinami. (Slovo “diskrétní” je zde míněno jako opak “spojitého”.) Dá se říci, že diskrétní matematika je oblastí matematiky, která je nejbližší informatice, a její poměrně nedávný prudký rozvoj byl do značné míry inspirován právě informatikou, obzvláště tou teoretickou.

Pro začátek úspěšného studia je velmi důležité probrat různé kombinatorické způsoby výběrů prvků z množiny, neboť tyto znalosti jsou přímo či nepřímo používány ve většině aplikací diskrétní matematiky. Jedná se především o výběry bez pořadí či výběry uspořádané a o princip skládání nezávislých výběrů. (Čtenáři se zajisté už s některými z nich prakticky setkali dříve.) I náš učební text je na těchto znalostech založen a očekává od čtenáře jejich dobré zvládnutí.

#### Cíle

Za hlavní cíl úvodní části považujeme správné pochopení principů různých typů kombinatorických výběrů – permutací, kombinací a variací, včetně těch s opakováním prvků. Dále je potřeba si zapamatovat základní vzorce pro počty různých typů výběrů.

#### 1.1 Čísla, operace a množiny

Úvodem si připomeneme známé matematické pojmy týkající se převážně přirozených čísel a konečných množin. Zároveň si zde zavedeme (pro ujednocení) některé základní konvence a značení používané dále v našem textu.

#### Přirozená čísla

Množina *přirozených čísel* je tvořena prvky  $\mathbb{N} = \{0, 1, 2, 3, 4, 5, 6, 7, \dots\}$ . (Pozor, všimněte si, že *přirozená čísla obsahují 0*.)

- Značení intervalu přirozených čísel  $[a, b] = \{a, a + 1, \dots, b - 1, b\}$ .
- *Celá část* reálného čísla  $x$  je značena následovně: dolní  $\lfloor x \rfloor$  a horní  $\lceil x \rceil$ .
- *Posloupnost*, obvykle značená  $(a_i)_{i=1}^n = (a_1, a_2, \dots, a_n)$ , je seřazením jakýchkoliv  $n$  prvků za sebou. (Prvky posloupnosti mohou být třeba čísla, ale také jakékoliv jiné objekty.) Pro referenci jsou prvky posloupnosti indexovány přirozenými čísly. V posloupnosti se, na rozdíl od množiny, prvky mohou volně opakovat.
- *Suma* posloupnosti čísel je zapsána

$$\sum_{i=1}^n a_i = a_1 + a_2 + \dots + a_{n-1} + a_n,$$

nebo také obecněji  $\sum_{i \in J} a_i = a_{i_1} + a_{i_2} + \dots + a_{i_n}$ , kde  $J = \{i_1, i_2, \dots, i_n\}$  je jakákoliv množina indexů.

- *Součin* posloupnosti  $n$  čísel je analogicky zapisován jako

$$\prod_{i=1}^n a_i = a_1 \cdot a_2 \cdot \dots \cdot a_{n-1} \cdot a_n.$$

**Komentář:** Pro ukázkou připojíme několik příkladů použití značení sum a součinů:

$$\sum_{i=1}^{10} i = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 = 55$$

$$\prod_{i=1}^5 (i+1) = 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 = 720$$

$$\sum_{i=1}^n \sum_{j=1}^n (i \cdot j) = \sum_{i=1}^n \left( i \cdot \sum_{j=1}^n j \right) = \left( \sum_{i=1}^n i \right) \cdot \left( \sum_{j=1}^n j \right) = \left( \frac{1}{2} n(n+1) \right)^2$$

Čtenář nechť si sám promyslí, co poslední uvedený zápis vlastně znamená, není v něm skryto nic složitého.

## Množiny a množinové operace

Opět si ve stručnosti připomeneme či zavedeme některá značení používaná v teorii množin.

- Obvykle značíme *množiny* velkými písmeny  $A, B, X, Y, \dots$  a jejich *prvky* malými písmeny  $a, b, x, y, \dots$ .  
Symbol  $\emptyset$  značí *prázdnou množinu*.
- $M = \{a, b, c, x\}$  je množina se čtyřmi prvky  $a, b, c, x$  a píšeme  $a \in M$ ,  $b \in M$ , ale  $d \notin M$ . Počet prvků (*mohutnost*) značíme  $|M| = 4$ .  
 $N = \{a, b, x\}$  je *podmnožinou*  $M$ , což píšeme jako  $N \subseteq M$ .
- *Sjednocení* množin značíme  $A \cup B$ , jejich *průnik*  $A \cap B$ .
- *Rozdíl* množin je definován  $A \setminus B = \{a \in A : a \notin B\}$ ,  
*symetrický rozdíl* je  $A \Delta B = (A \setminus B) \cup (B \setminus A)$  (jako “XOR”).
- Rozšířené značení sjednocení  $\bigcup_{i=1}^n X_i$  a průniku  $\bigcap_{i=1}^n X_i$  je analogické jako u sumy.

**Komentář:** Pro názornou ukázkou si zvolme například množiny

$$A = \{a, b, c, d\}, \quad B = \{b, c, e, f\}.$$

Pak sjednocením těchto dvou množin je  $A \cup B = \{a, b, c, d, e, f\}$  a průnikem je  $A \cap B = \{b, c\}$ .  
Výsledkem rozdílu je  $A \setminus B = \{a, d\}$  a symetrického rozdílu  $A \Delta B = \{a, d, e, f\}$ .

V další ukázce nechť je  $S$  množinou všech sudých přirozených čísel a  $P$  je množinou všech prvočísel. Pak, jak je dobře známo, v průniku  $S \cap P$  je jediné číslo 2. Rozdíl  $P \setminus S$  obsahuje všechna lichá prvočísla, kdežto obrácený rozdíl  $S \setminus P$  obsahuje všechna sudá přirozená čísla mimo 2.

- *Potenční množina* (množina všech podmnožin) množiny  $X$  je

$$2^X = \{A : A \subseteq X\}.$$

- *Množinový systém* nad  $X$  (což je vhodněji znějící název místo “množina množin”) je jakákoliv podmnožina potenční množiny  $\mathcal{T} \subseteq 2^X$ .
- *Kartézský součin* dvou množin je množinou všech uspořádaných dvojic vybraných po složkách z těchto množin, tj.

$$A \times B = \{(a, b) : a \in A, b \in B\}.$$

- *Kartézská mocnina* je definována rekurzivně

$$A^n = \underbrace{A \times A \times \dots \times A}_n,$$

kde  $A^1 = A$  a  $A^0 = \{\emptyset\}$ .

**Komentář:** Podívejme se například na potenční množinu dvouprvkové množiny  $\{a, b\}$ . Ta obsahuje prázdnou množinu, dvě jednoprvkové množiny s těmito prvky a celou dvouprvkovou množinu:

$$2^{\{a,b\}} = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$$

Pro počet prvků potenční množiny platí přirozený vzorec:

$$|2^X| = 2^{|X|}$$

V dalším příkladě si ukažme jednoduchý kartézský součin dvou dvouprvkových množin:

$$\{a, b\} \times \{c, d\} = \{(a, c), (a, d), (b, c), (b, d)\}$$

Počty prvků v kartézském součinu a v kartézské mocnině splňují

$$|A \times B| = |A| \cdot |B|, \quad |A^n| = |A|^n.$$

## Úlohy k řešení

(1.1.1) Sečtěte sumu  $\sum_{i=1}^6 i^2$ .

(1.1.2) Vynásobte  $\prod_{j=9}^{11} j$ .

(1.1.3) Kolik je  $\lfloor 3.3 \rfloor$ ?

(1.1.4) Kolik je  $\lceil -3.3 \rceil$ ?

(1.1.5) Kolik je  $\lceil 3.6 \rceil$ ?

(1.1.6) Kolik je  $\lceil -3.6 \rceil$ ?

(1.1.7) Kolik prvků má potenční množina množiny  $\{1, 2, 3, 4\}$ ?

(1.1.8) Kolik prvků je v kartézském součinu  $\{1, 2, 3, 4\} \times \{5, 6, 7\}$ ?

\* (1.1.9) Jak vyjádříte klasické zaokrouhlení pomocí  $\lfloor \cdot \rfloor$ ?

## 1.2 Výběry a seřazení prvků

V tomto místě začneme přehledem tří základních kombinatorických způsobů výběru prvků bez opakování. Představme si množinu  $n$  prvků  $X$ , ze které hodláme vybírat. V prvé řadě můžeme vzít rovnou všechny prvky a dívat se na jejich možná seřazení. Tento pohled je formalizován následující definicí:



**Definice 1.1. *Permutace***  $n$ -prvkové množiny  $X$  je *uspořádání všech prvků* z  $X$  do jedné posloupnosti (délky  $n$ ) bez opakování prvků. Počet všech permutací  $n$ -prvkové množiny je dán vzorcem

$$P(n) = n! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot \dots \cdot (n-1) \cdot n \quad (0! = 1).$$

Jindy nás pořadí vybíraných prvků z  $X$  nemusí vůbec zajímat (případně pořadí může být již dáno, tj. *nevybírání se pořadí*), ale soustředíme se na to, které prvky se do výběru dostanou a které ne. To je zachyceno ve druhé definici:

**Definice 1.2. *Kombinace*** na množině  $X$  je *neuspořádaný výběr* některých prvků z  $X$  bez opakování, tj. výběr podmnožiny  $X$ . Počet  $k$ -prvkových kombinací na  $n$ -prvkové množině je dán vzorcem

$$C(n, k) = \binom{n}{k} = \frac{n!}{k! \cdot (n-k)!}.$$

A nakonec nás může zajímat obojí – jak to, které prvky jsou vybrány, tak i to, jak je jejich výběr seřazen.

**Definice 1.3. *Variace*** na množině  $X$  je *uspořádaný výběr* některých prvků z  $X$  bez opakování, tj. podposloupnost množ.  $X$  bez opakování.

Počet  $k$ -prvkových variací na  $n$ -prvkové množině je dán vzorcem

$$V(n, k) = \frac{n!}{(n-k)!}.$$

**Komentář:** Pro ilustraci se podívejme na příklad – máme klub 15 (stejných) sportovců, ze kterých bychom měli poslat 3-členné družstvo na závody.

- Vybíráme-li 3-členné družstvo z 15 sportovců, jedná se o *kombinace*, tj. počet možností je  $\binom{15}{3} = 455$ . (Na jejich seřazení nám v tuto chvíli nezáleží.)
- Chceme-li vybrané 3-členné družstvo seřadit na slavnostní nástup, jedná se o *permutaci*, tj. máme celkem  $3! = 6$  možností.
- Představme si však, že rovnou hodláme vybrat z těchto 15 sportovců toho prvního, druhého a třetího do reprezentačního družstva. Pak se jedná o *variace*, tj. máme celkem  $15!/12! = 2730$  možností. Všimněte si dobře, že  $455 \cdot 6 = 2730$ ; to je přirozené, neboť sportovce můžeme nejprve vybrat jako kombinaci a pak je seřadit.

**Poznámka** o výpočtu kombinačních čísel:

Definiční vztah  $\binom{n}{k} = \frac{n!}{k! \cdot (n-k)!}$  se pro praktický výpočet kombinačního čísla vůbec nehodí, neboť funkce faktoriál roste tak rychle, že brzy přesáhne možnosti vaší kalkulačky. Místo toho obvykle používáme zde popsáný postup. Předpokládejme, že  $k \leq \frac{n}{2}$ , jinak vezměme raději číslo  $\binom{n}{n-k} = \binom{n}{k}$ . Pak si definiční vztah upravme:

$$\binom{n}{k} = \frac{n!}{k! \cdot (n-k)!} = \frac{n(n-1)(n-2)\dots(n-k+1)}{k!}$$

Například

$$\binom{100}{3} = \frac{100 \cdot 99 \cdot 98}{3 \cdot 2} = 50 \cdot 33 \cdot 98.$$

Pro výpočet variací používáme obdobnou úpravu

## Řešení složených výběrů

Při řešení praktických kombinatorických problémů však ne vždy (dokonce málokdy) jsme v situaci, kdy řešení přímo padne do jedné z uvedených tří Definic 1.1–1.3. Dalšími možnostmi jsou kombinace či permutace s opakováním, které popíšeme v příští části. Ale často se v problémech jedná o *složené výběry*, sestávající z několika dílčích podvýběrů. Základní (a přirozený) postup řešení takovýchto složených úloh je popsán následovně:

### Metoda 1.4. *Princip nezávislých výběrů*

*V situaci složeného výběru, kdy není vůbec žádná vazba každého jednoho částečného výběru (podvýběru) k ostatním podvýběrům, stačí k získání celkového výsledku mezi sebou vynásobit počty možností jednotlivých podvýběrů.*

**Komentář:** Princip nezávislých výběrů bude nejlépe ilustrován několika praktickými ukázkami.

- Vybíráme-li delegaci 2 studentů a 1 studentky ze skupiny 20 studentů a 3 studentek, pak můžeme nezávisle vybrat nejprve studenty a pak studentku (či naopak). Počet výběrů je vynásobením  $\binom{20}{2} \cdot \binom{3}{1} = 570$ .
- Vybíráme-li z 15 hráčů (například minifotbalu) družstvo 3 + brankář, pak ale podvýběr trojice a podvýběr brankáře **nezávislé nejsou**. (Striktně řečeno, jakmile zvolíme trojici hráčů do pole, žádný z nich již nemůže být brankářem.) Přesto i zde *s opatrností násobit lze*, pokud každý z oněch 15 hráčů může být stejně dobře brankářem – (pod)vybereme trojici a na brankáře v každém případě zbude 12 voleb. Celkem tak je  $\binom{15}{3} \cdot 12 = 5460$  různých možností výběru.

Další možnost řešení některých složených výběrů spočívá v tom, že požadovaný výběr ještě dále “zjemníme” tak, abychom dostali jiný výběr, který již dokážeme spočítat některou z jiných metod. Počet možností původního výběru pak získáme vydělením.

### Metoda 1.5. *Dvojího počítání*

*Nechť každý případ nějakého (složeného) výběru lze dále rozlišit (zjemnit) na stejný počet  $l$  zjemněných možností. Dále nechť získaný zjemněný výběr má celkem  $m$  různých možností (které jsme schopni spočítat). Potom počet všech možností původního výběru je dán podílem  $m/l$ .*

**Komentář:** Opět je nejlepší popsanou metodu dvojího počítání ilustrovat názornou ukázkou použití. (Později budou uvedeny i jiné příklady této metody, především v Lekci 3.)

Vraťme se ke druhé ukázce pro Metodu 1.4 – výběru družstva 3 + brankář z 15 hráčů.

- Dále si představme, že chceme ještě navíc takovéto družstvo seřadit na slavnostní nástup tak, aby brankář byl první.
- Pokud je počet možných výběrů našeho družstva  $x$ , pak v každém případě musíme postavit brankáře prvního, ale zbylé tři hráče můžeme seřadit  $3! = 6$  způsoby za ním. Celkem tedy získáme  $x \cdot 1 \cdot 3!$  seřazení na nástup.
- Na druhé straně si můžeme situaci představit tak, že rovnou vybíráme variace 4 hráčů ze všech 15 a že prvního z nich prohlásíme za brankáře. Počet možností je dle vzorce  $V(15, 4) = \frac{15!}{11!}$ .
- Oba pohledy jsou však ekvivalentní, takže počty musí souhlasit:

$$x \cdot 1 \cdot 3! = V(15, 4) = \frac{15!}{11!}$$

Po vydělení:

$$x = \frac{15!}{11! \cdot 3!} = 5460$$

(Určitě není náhodou, že výsledek vyšel stejný, že?)

### Úlohy k řešení

- (1.2.1) Upravte na jednoduchý součin  $\binom{6^n}{3}$ .
- (1.2.2) Kolika způsoby seřadíme šestici studentů do řady?
- (1.2.3) Kolika způsoby vyberete neuspořádanou dvojici z 21 studentů?
- (1.2.4) Kolika způsoby vyberete uspořádanou dvojici z 21 studentů?
- (1.2.5) Z osudí 49 čísel táhneme jedno, vrátíme jej a táhneme druhé. Kolik různých výsledků může vyjít?
- \*(1.2.6) Co když v předchozím výběru dvou čísel zapomeneme jejich pořadí, kolik různých výsledků získáme?
- \*(1.2.7) Hokejový trenér má k dispozici 13 útočníků a 9 obránců. Kolika způsoby vybere pětku hráčů? (2 obránci + 3 útočníci)

### 1.3 Výběry s opakováním

Doposud jsme se zabývali výběry prvků, které se nemohou opakovat. Nemožnost opakování prvků je přirozenou podmínkou například při výběrech skupin lidí, kdežto v jiných situacích a výběrech jsou opakování prvků žádoucí. Tak je to třeba v situaci, kdy vybíráme slova složená z daných písmen-prvků. Následující příklad uvádí tzv. permutace s opakováním.

**Příklad 1.6.** Kolika různými způsoby můžeme seřadit do posloupnosti písmena sousloví DISKRETNIMATEMATIKA ?

Pokud bychom si výskyty stejných písmen barevně rozlišili, např. DISKRETNIMATEMATIKA, pak je počet seřazení roven počtu permutací z 19 znaků, ale my neumíme různá pořadí písmene A či I atd. odlišit.

Proto použijme metodu dvojího počítání (Metoda 1.5). Necht  $x$  je neznámý počet různých seřazení našich písmen. V každém takovém seřazení si dodatečně obarvíme stejná písmena různými barvami a sestrojme  $3!$  permutací písmen I,  $2!$  permutací písmen K,  $2!$  permutací písmen E,  $3!$  permutací písmen T,  $2!$  permutací písmen M a  $3!$  permutací písmen A. Celkem tak získáme všech  $19!$  permutací z rozlišených 19 znaků, a proto

$$(3!)^3 \cdot (2!)^3 \cdot x = 19!,$$

Po vydělení:

$$x = \frac{19!}{(3!)^3 \cdot (2!)^3} = \frac{19!}{3! \cdot 3! \cdot 3! \cdot 2! \cdot 2! \cdot 2! \cdot 1! \cdot 1! \cdot 1! \cdot 1!}.$$

□

Použití metody dvojího počítání v tomto příkladě lze jednoduše zobecnit na obecné permutace s opakováním, jak dokládá následující definice.

**Definice 1.7.** *Permutace s opakováním* prvků množiny  $X$  je seřazení (*uspořádání*) prvků z  $X$  do posloupnosti takové, ve které se každý prvek  $X$  opakuje předepsaný počet krát. (Jinými slovy, je to seřazení předepsaných počtů identických kopií prvků z  $X$ .)

Počet všech permutací s opakováním z  $k$  prvků, kde  $i$ -tý prvek se opakuje  $m_i$ -krát pro  $i = 1, \dots, k$ , je

$$\frac{(m_1 + m_2 + \dots + m_k)!}{m_1! \cdot m_2! \cdot \dots \cdot m_k!}.$$

**Komentář:** V Příkladě 1.6 jsme měli 10 různých písmen, která se opakovala následující počty krát:  $m_D = 1$ ,  $m_I = 3$ ,  $m_S = 1$ ,  $m_K = 2$ ,  $m_R = 1$ ,  $m_E = 2$ ,  $m_T = 3$ ,  $m_N = 1$ ,  $m_M = 2$  a  $m_A = 3$ . Takže výsledek příkladu je stejný, jako bychom jej dostali dosazením do vzorce v Definici 1.7.

**Poznámka:** Pokud uvažujeme permutace s opakováním z 2 prvků, kde se první prvek opakuje  $k$ -krát a druhý  $(n - k)$ -krát, dostáváme vztah pro jejich počet

$$\frac{(k + n - k)!}{k! \cdot (n - k)!} = \binom{n}{k} = C(n, k).$$

Jedná se tedy o tentýž počet jako  $k$ -prvkových kombinací z  $n$  prvků. To není náhoda, protože na výběr kombinace se můžeme dívat jako na seřazení  $k$  černých a  $(n - k)$  bílých značek, kde černá značka znamená, že prvek vybereme do podmnožiny, a bílá naopak ne.

**Poznámka:** Na permutace s opakováním se lze dívat ještě jedním pohledem. V kombinatorice se také definuje pojem “multimnožiny”, která je analogická množině, ale jednotlivé prvky se v ní mohou vyskytovat v několika identických kopiích. V tomto pohledu pak permutace s opakováním je jako běžná permutace, ale nad multimnožinou místo množiny.

Další zajímavý příklad uvádí tzv. kombinace s opakováním.

**Příklad 1.8.** *Kolika různými způsoby můžeme vybrat 6 kuliček ze tří barev, když počet kuliček stejné barvy není omezený?*

Zde hledáme *neuspořádaný výběr*. Představme si proto, že kuličky následně (jednoznačně) seřadíme podle barev v pořadí červená, modrá, zelená. Například  $\bullet, \bullet, \bullet, \bullet, \bullet, \bullet$  napíšeme jako

$$\bullet \mid \bullet \bullet \mid \bullet \bullet \bullet,$$

kde znak  $\mid$  odděluje jednotlivé barvy. Nyní však vidíme, že barvy můžeme zapomenout, protože jsou už určené pozicemi oddělovačů  $\mid$ , a můžeme psát

$$\bullet \mid \bullet \bullet \mid \bullet \bullet \bullet.$$

Takže na určení výběru kuliček stačí vybrat 2 pozice oddělovačů z  $6 + 2 = 8$  možností (neuspořádaně). To dává výsledek

$$C(8, 2) = \binom{8}{2}.$$

(Uvědomme si, že skutečně vybíráme oddělovače z osmi pozic, ne ze sedmi mezer mezi kuličkami, protože v jedné mezeře mohou být oba oddělovače najednou.)  $\square$

Opět můžeme uvedené řešení snadno zobecnit na všechny kombinace s opakováním, jako v následující definici:

**Definice 1.9.** *Kombinace s opakováním* z množiny  $X$  je výběr *neuspořádaného “seskupení”* (multimnožiny) prvků z  $X$ , přičemž se prvky výběru mohou *opakovat v libovolném počtu* (identických kopií).

Počet všech  $k$ -prvkových kombinací s opakováním z  $n$  možností je

$$C^*(n, k) = \binom{n + k - 1}{n - 1}.$$

Vzorec pro kombinace s opakováním má mnoho šikovných a zajímavých aplikací. Pro ilustraci uvedeme následující příklady.

**Příklad 1.10.** *Kolika různými způsoby můžeme zapsat přirozené číslo  $k$  jako součet  $n$  přirozených sčítanců?*

Mějme součet

$$k = x_1 + x_2 + \dots + x_n.$$

V analogii s Příkladem 1.8 o výběrech barevných kuliček můžeme vidět číslo  $x_1$  jako počet červených kuliček,  $x_2$  jako počet modrých, atd. Takže součet si lze představit jako  $k$ -prvkovou kombinaci s opakováním z  $n$  prvků, kde  $i$ -tý prvek se opakuje právě  $x_i$  krát. Proto je počet takových zápisů na součty dán už známým vztahem

$$C^*(n, k) = \binom{n+k-1}{n-1}.$$

Pro bližší vysvětlení: Přestože sčítance v součtu  $x_1 + x_2 + \dots + x_n$  jsou uspořádány svými indexy, nejedná se zde v žádném případě o uspořádaný výběr, neboť pořadí sčítanců je dáno a hraje roli jenom pro odlišení sčítanců od sebe, ale nevybírání se!  $\square$

### Úlohy k řešení

- (1.3.1) Kolika různými způsoby seřadíme do posloupnosti písmena slova ANAKONDA?
- (1.3.2) Automobilové rallye se účastní vozy čtyř značek, tři vozy od každé značky. Kolik různých výsledných pořadí může být, pokud nerozlišujeme vozy téže značky?
- (1.3.3) Kolika způsoby rozložíme číslo 7 na součet tří přirozených sčítanců? (Na jejich pořadí záleží.)
- \* (1.3.4) Kolika různými způsoby můžeme nabarvit 6 identických bílých kuliček jednou ze čtyř barev? (Každá kulička musí být nabarvena.)
- \* (1.3.5) Kolika různými způsoby můžeme nabarvit 6 očíslovaných kuliček jednou ze čtyř barev? (Každá kulička musí být nabarvena a čísla jsou pod barvou stále vidět – jsou vyrytá.)

### Rozšiřující studium

Podrobnější zopakování pojmů o přirozených číslech a množinách čtenář najde v doporučené učebnici Matouška a Nešetřila [5, Oddíl 1.2], nebo v jiných učebnicích diskrétní matematiky či algebry. O kombinatorických výběrech a jejich počítání si zaujatý čtenář může přečíst více v [5, Kapitola 2]. Specificky metodě dvojího počítání je věnována celá [5, Kapitola 6].

## 1.4 Cvičení: Kombinatorické výběry

**Příklad 1.11.** Kolik podmnožin liché velikosti lze vybrat z 33 prvků?

Nechť množina  $X$  má 33 prvků a vybíráme lichou podmnožinu  $L \subseteq X$ . Podívejme se, že doplněk  $X \setminus L$  má pak sudou velikost a naopak. To znamená, že ke každé liché podmnožině je jednoznačně přiřazena jiná sudá podmnožina, a proto je lichých podmnožin právě polovina všech podmnožin. To jest celkem  $\frac{1}{2} \cdot 2^{33} = 2^{32}$  lichých podmnožin.  $\square$

**Příklad 1.12.** Mějme k dispozici 12 hráčů, mezi nimiž je 5 dobrých útočníků. Kolika způsoby můžeme sestavit 4-členné družstvo, ve kterém bude alespoň jeden dobrý útočník?

Na první pohled by se mohlo zdát, že nejprve vybereme jednoho dobrého útočníka a pak dobereme tři zbylé hráče. Tato úvaha má však jednu závažnou chybu – pokud vybrané družstvo má třeba dva dobré útočníky, pak jej započítáme dvakrát, za každého jednoho dobrého útočníka zvlášť.

Proto budeme postupovat jinak. Všechných výběrů 4-členných družstev je  $\binom{12}{4}$ . Mezi nimi jsou i ty špatné výběry, které neobsahují žádného dobrého útočníka. Takových špatných

výběrů je  $\binom{12-5}{4} = \binom{7}{4}$ , vybíráme totiž čtyři hráče z těch zbylých. Špatné výběry od všech jednoduše odečteme, takže výsledek je

$$\binom{12}{4} - \binom{7}{4}.$$

□

**Příklad 1.13.** *Kolika různými způsoby můžeme zapsat přirozené číslo  $k$  jako součet  $n$  celých kladných sčítanců?*

Mějme součet s kladnými sčítanci

$$k = x_1 + x_2 + \dots + x_n.$$

To lze ekvivalentně zapsat jako

$$k - n = (x_1 - 1) + (x_2 - 1) + \dots + (x_n - 1),$$

kde  $y_i = x_i - 1$  jsou přirozená čísla. Jinak řečeno, zapisujeme číslo  $k - n$  jako součet  $n$  přirozených sčítanců. Dále už postupujeme jako v Příkladě 1.10, tedy celkový počet způsobů je

$$C^*(n, k - n) = \binom{n + k - n - 1}{n - 1} = \binom{k - 1}{n - 1}.$$

□

### Úlohy k řešení

**(1.4.1)** *Kolik různých součtů může padnout pokud hodíme najednou třemi kostkami s počty stěn 6, 10 a 12? (Kostky jsou číslovány od 1 do počtu stěn.)*

Návod: Ptáme se na hodnoty součtů, co padnou, takže třeba  $2 + 3 + 7$  je totéž co  $1 + 2 + 9$ .

**(1.4.2)** *V tenisovém klubu se sešlo 5 hráčů. V kolika dvojicích si mohou zahrát dvouhru?*

**(1.4.3)** *V tenisovém klubu se sešlo 9 hráčů. Kolik různých sestav na čtyřhru mohou vytvořit? Zdůvodněte svým úsudkem.*

**(1.4.4)** *Kolika způsoby můžeme zamíchat 32 karet, aby byly v neklesajícím pořadí, přičemž se neohlížíme na jejich barvy? (Tj. nebude král před dárou a podobně.)*

\***(1.4.5)** *Kolika způsoby můžeme nabarvit některé z 10 bílých kuliček jednou ze čtyř barev?*

\***(1.4.6)** *Kolika způsoby lze psát číslo  $\ell$  jako součet  $n$  sčítanců s hodnotami 1 nebo 2?*

Návod: Pozor, toto nelze řešit jako Příklad 1.10. Ve skutečnosti tento příklad vede na běžné kombinace bez opakování – vybíráme ty ze sčítanců, které jsou hodnoty 2.

## 2 Diskrétní pravděpodobnost

### Úvod

Jedním ze základních historických motivačních zdrojů diskrétní matematiky, konkrétně klasické kombinatoriky, jsou hazardní hry. Hráči se odjakživa zamýšleli nad tím, jakou mají šanci hodit určitou kombinaci kostek nebo dostat do ruky jisté karty. Tyto otázky motivovaly jak zkoumání různých způsobů kombinatorických výběrů, tak i matematickou formalizaci důležitého filozofického pojmu pravděpodobnosti.

Zabývat se zde budeme pouze (diskrétními) událostmi, ve kterých může nastat jedna z konečně mnoha izolovaných možností. Zhruba řečeno, náhodná událost je v diskrétní pravděpodobnosti modelována výběrem určitých jevů z prostoru všech možných jevů a její hodnota pravděpodobnosti je rovna relativní četnosti vybraných jevů vůči všem možným.

Učivo plynule navazuje na běžné středoškolské úlohy o hodech mincí, kostkou, či míchání karet, a dává jim pevné formální základy.

## Cíle

Hlavním cílem lekce je pochopení modelu diskrétní pravděpodobnosti pomocí konečného pravděpodobnostního prostoru a relativní četnosti jevů na něm. Čtenář se naučí, jak používat model diskrétní pravděpodobnosti a jeho matematický aparát (jako sjednocení a průnik jevů, nezávislost, střední hodnotu, atd.) na řešení praktických problémů, především těch s náhodnými výběry.

## 2.1 Motivační příklady

Před uvedením formální definice pravděpodobnostního prostoru začneme několika příklady pro osvětlení problematiky. V každém z nich se jedná o dobře známý náhodný proces s jedním z několika (předem známých) možných výsledků.

- *Hod mincí* – má 2 možné výsledky hlava / orel, neboli  $1/0$ . Každý padá se zhruba stejnou četností (říkáme “s pravděpodobností  $\frac{1}{2}$ ”).
- *Hod kostkou* – má 6 možných výsledků 1, 2, 3, 4, 5, 6. Opět každý padá se stejnou četností (říkáme “s pravděpodobností  $\frac{1}{6}$ ”).
- *Zamíchání karet* – předpokládáme, že každé možné zamíchání karet lze asi stejně dobře očekávat. Zde je však všech možností nesrovnatelně více –  $32! \doteq 2.6 \cdot 10^{35}$ , což vůbec nejsme schopni ani vypsát.

*Poznámka:* Tento přirozený příklad zároveň přináší zajímavou filozofickou otázku: Jak můžeme tvrdit, že každé možné pořadí karet nastane stejně pravděpodobně, když se patrně za celou dobu existence lidstva a karet skutečně objeví jen nepatrný zlomek všech zamíchání? Toto lze seriózně rozřešit pouze použitím přesného formálního matematického modelu pravděpodobnosti.

- *Tah sportky* – očekáváme, že každé příští tažené číslo bude “stejně pravděpodobné” ze všech čísel zbylých v osudí. Celý tah je však ve výsledku neuspořádaným výběrem, takže počet všech možností je  $\binom{49}{7} = 85900584$ . I to je příliš vysoké číslo pro snadnou představu.

Často tak při našem pohledu na náhodné události přirozeně vycházíme z toho, že jisté různé možnosti si jsou “ekvivalentní” co do četnosti jejich výskytu (například proto, že fyzický postup výběru by mezi nimi neměl umět rozlišit za předpokladu poctivosti). Na tomto předpokladu pak stavíme naše “*očekávání pravděpodobnosti*”.

## 2.2 Konečný pravděpodobnostní prostor

V matematice chceme nějak modelovat chování “*náhody*”: Náhodu přitom subjektivně vnímáme jako naše očekávání, zda nějaký jev nastane. Toto očekávání “měříme” našimi zkušenostmi s předchozí *relativní četností opakování* zkoumaného jevu. Proto je nasnadě se na pravděpodobnost matematicky dívat jako na číslo vyjadřující relativní četnost opakování jevu při velmi velkém množství pokusů. Seriózní formální základ tomuto pohledu poskytuje následující důležitá definice.

### Definice 2.1. *Konečný pravděpodobnostní prostor*

je formálně dvojice  $(\Omega, P)$ , kde  $\Omega$  je konečná množina *elementárních jevů* a  $P$  je *funkce pravděpodobnosti*, která podmnožinám  $\Omega$  přiřazuje reálné hodnoty z intervalu  $[0, 1]$  a

splňuje

- $P(\emptyset) = 0$ ,  $P(\Omega) = 1$ ,
- $P(A \cup B) = P(A) + P(B)$  pro disjunktní  $A, B \subseteq \Omega$ .

*Jev* je libovolná podmnožina  $A \subseteq \Omega$  a jeho *pravděpodobnost* je  $P(A)$ .

Čtenář nechť si dobře uvědomí významy pojmů *jev* a *elementární jev* a jejich vzájemný vztah. Elementární jev je jeden (kterýkoliv) prvek prostoru  $\Omega$ , neboli také příslušná jednoprvková podmnožina. Jev v obecnosti je *jakákoliv* podmnožina prostoru  $\Omega$ , tedy se skládá z několika elementárních jevů.

**Definice:** Jevy  $A, B$  jsou *disjunktní*, pokud nemohou nastat zároveň, tj. pokud  $A \cap B = \emptyset$ . (Pravděpodobnosti disjunktních jevů se dle Definice 2.1 sčítají.)

**Poznámka:** Pravděpodobnostní prostor je plně určený množinou  $\Omega$  a pravděpodobnostmi elementárních jevů – prvků  $\Omega$ . Pro  $A = \{a_1, \dots, a_k\} \subseteq \Omega$  totiž z definice platí

$$P(A) = P(a_1) + P(a_2) + \dots + P(a_k).$$

Nejčastější a nejpřirozenější funkcí pravděpodobnosti v praktických příkladech je následující:

**Definice:** *Uniformní pravděpodobnost* je prostoru  $\Omega$  přiřazena funkcí

$$P(A) = |A| / |\Omega|,$$

tj. každý elementární jev je stejně pravděpodobný a pravděpodobnost každého obecného jevu je dána jeho relativní velikostí vzhledem k celému prostoru  $\Omega$ .

**Komentář:** Vraťme se k příkladům náhodných událostí z Oddílu 2.1 a ukažme si, jak jsou ony popsány pravděpodobnostními prostory ve smyslu Definice 2.1.

- Při hodu mincí:  $\Omega = \{0, 1\}$  a  $P(0) = P(1) = \frac{1}{2}$ .
- Při hodu kostkou:  $\Omega = \{1, 2, 3, 4, 5, 6\}$  a  $P(1) = \dots = P(6) = \frac{1}{6}$ .  
Například jev “padlo sudé číslo” je tvořen podmnožinou  $\{2, 4, 6\}$ .
- Při zamíchání karet:  $\Omega$  je tvořena všemi  $32!$  permutacemi z 32 karet a každá permutace má stejnou pravděpodobnost  $\frac{1}{32!}$ .  
Například jev “první je eso” je tvořen podmnožinou všech těch permutací, co mají jako první kartu jedno z es.
- Při tahu sportky:  $\Omega$  je tvořena všemi 7-prvkovými kombinacemi z 49 čísel a každá kombinace má pravděpodobnost  $1/\binom{49}{7}$ .

V jiných jednoduchých příkladech definujeme pravděpodobnostní prostor analogicky, ale jsou i složitější náhodné procesy, kde se již musíme dobře zamyslet nad tím, co jsou elementární jevy a jakou mají pravděpodobnost.

**Příklad 2.2.** Představme si náhodný proces, kdy hodíme dvěma šestistěnnými kostkami a zjišťujeme jejich výsledný součet. Popište jej pravděpodobnostním prostorem, ve kterém elementárními jevy jsou jednotlivé součty.

Množina všech možných součtů dvou kostek je zřejmě  $\Omega = \{2, 3, \dots, 12\}$ . Dobře se ale zamysleme nad pravděpodobnostmi těchto elementárních jevů. Je poměrně zřejmé, že součet 2, který lze získat jediným způsobem  $1 + 1$ , bude padat méně často než součet 7, který lze získat šesti různými způsoby. Celkem je  $6 \cdot 6 = 36$  možností hodů dvou kostek, z toho, jak jsme si všimli, součet 7 padne šesti způsoby, součet 6 pěti způsoby, atd. Užitím úvahy o poměrném počtu způsobů tak získáme funkci pravděpodobnosti

$$\begin{aligned} P(2) &= P(12) = \frac{1}{36}, \\ P(3) &= P(11) = \frac{2}{36} = \frac{1}{18}, \end{aligned}$$



$$\begin{aligned}
P(4) &= P(10) = \frac{3}{36} = \frac{1}{12}, \\
P(5) &= P(9) = \frac{4}{36} = \frac{1}{9}, \\
P(6) &= P(8) = \frac{5}{36}, \\
P(7) &= \frac{6}{36} = \frac{1}{6}.
\end{aligned}$$

Ve skutečnosti je však lepší a přímočařejší použít jiný model s uniformní pravděpodobností, který je popsán v příštím příkladě.  $\square$

**Příklad 2.3.** Popište pravděpodobnostním modelem Příklad 2.2 za použití prostoru s uniformní pravděpodobností.

Jak jsme při řešení Příkladu 2.2 viděli, naše snaha prohlásit za elementární jevy při hodu dvou kostek jednotlivé součty byla dosti křečovitá a celé řešení přirozeně inklinovalo k modelu, kde elementárními jevy jsou jednotlivé dvojice hozených čísel. Takže mějme pravděpodobnostní prostor  $\Omega' = [1, 6]^2$  (druhá kartézská mocnina množiny  $\{1, \dots, 6\}$ ). Jevy jednotlivých součtů potom jsou

$$\begin{aligned}
S_1 &= \emptyset, \\
S_2 &= \{(1, 1)\}, \\
S_3 &= \{(1, 2), (2, 1)\}, \\
&\dots \\
S_7 &= \{(1, 6), (2, 5), (3, 4), (4, 3), (5, 2), (6, 1)\}, \\
S_8 &= \{(2, 6), (3, 5), (4, 4), (5, 3), (6, 2)\}, \\
&\dots \\
S_{12} &= \{(6, 6)\}.
\end{aligned}$$

Jelikož pravděpodobnost jedné dvojice hozených čísel, tj. jednoho elementárního jevu v  $\Omega'$ , je uniformní  $\frac{1}{36}$ , snadno z tohoto popisu odvodíme stejné pravděpodobnosti jako v Příkladě 2.2. Toto řešení je přitom snazší a matematicky přesnější.  $\square$

## Úlohy k řešení

(2.2.1) Jaká je pravděpodobnost, že na kostce padne liché číslo?

(2.2.2) Hodíme dvěma kostkami. Jaká je pravděpodobnost, že padne součin 12?

(2.2.3) Jaká je pravděpodobnost, že po zamíchání karet bude třetí karta shora eso?

## 2.3 Nezávislost jevů

Nezávislost pravděpodobnostních jevů intuitivně znamená, že pravděpodobnost toho, že nastane druhý z nich, není nijak ovlivněna skutečným výsledkem prvního jevu, a naopak. Tento pojem je principiálně dosti podobný *nezávislým výběrům* z Lekce 1.

**Komentář:** Začneme několika jednoduchými ukázkami.

*Nezávislé jevy:* dva hody toutéž kostkou za sebou, hod kostkou a současné zamíchání karet, dva různé tahy sportky.

*Závislé jevy:* vrchní a spodní číslo padlé při jednom hodu kostky, výběr první a druhé karty ze zamíchaného balíčku, dvě akumulacní kurzové sázky obsahující stejné utkání.

Jak však nezávislost jevů definujeme formálně? Jedná se totiž spíše o filozofický pojem, který neumíme přímo matematicky postihnout, proto pro definici nezávislosti použijeme *součinnové pravidlo* (obdobné pravidlu pro násobení nezávislých výběrů), které je vlastně důsledkem intuitivního významu nezávislosti.

**Definice 2.4.** *Nezávislé jevy*  $A, B$

jsou takové, pro které platí  $P(A \cap B) = P(A) \cdot P(B)$ .

Nezávislost dvou jevů  $A, B$  lze ještě vyslovit následujícím způsobem. Jev  $B$  je nezávislý na  $A$ , pokud pravděpodobnost  $B$  při nastálém jevu  $A$  je stále stejná jako pravděpodobnost jevu  $B$  obecně, tedy úměrou

$$\frac{P(A \cap B)}{P(A)} = \frac{P(B)}{P(\Omega)}.$$

Jelikož  $P(\Omega) = 1$ , uvedený vztah je ekvivalentní definičnímu  $P(A \cap B) = P(A) \cdot P(B)$ .

**Poznámka:** Uvedená definice nezávislosti jevů si zaslouhuje bližší osvětlení, neboť je poněkud netradiční. Obvykle matematická definice pojmu obsahuje snadno ověřitelné podmínky, a pak následují tvrzení přisuzující definovanému pojmu různé užitečné vlastnosti. Zde je tomu však naopak, ona užitečná vlastnost, totiž pravidlo násobení pravděpodobností nezávislých výběrů, je přímo podmínkou v Definicí 2.4. Jak jej tedy použijeme při řešení příkladů?

Nejčastěji je to tak, že nezávislost dvou jevů vyplývá z kontextu situace, třeba se jedná o náhodné procesy bez jakékoliv fyzické vazby. V příkladě pak nezávislost takto zdůvodníme a součinné pravidlo využijeme při výpočtu.

**Poznámka:** Pravděpodobnosti dané vzorcem  $\frac{P(A \cap B)}{P(A)}$  se také říká *podmíněná pravděpodobnost* jevu  $B$  za předpokladu, že nastal jev  $A$ . Značí se  $P(B|A)$ .

**Příklad 2.5.** *Hodíme dvěma stejnými kostkami najednou. Jaká je pravděpodobnost, že nám padne 4 a 5?*

Hody obou kostek jsou nezávislé, neboť v poctivé situaci mezi nimi není fyzická vazba. Pravděpodobnost, že na první kostce padne 4, je  $\frac{1}{6}$ , obdobně je  $\frac{1}{6}$ , že na druhé padne 5. Obě možnosti najednou padnou s pravděpodobností po vynásobení  $\frac{1}{6} \cdot \frac{1}{6} = \frac{1}{36}$ .

Musíme si však dát pozor na to, že kostky jsou stejné a my nemáme určeno, na které padne 4. Proto se úloha vlastně rozpadá na dva disjunktní jevy *na první padne 4 a na druhé 5 a jev na první 5 a na druhé 4*, a proto výsledná pravděpodobnost je součtem obou možností  $P = \frac{1}{36} + \frac{1}{36} = \frac{1}{18}$ .  $\square$

## Úlohy k řešení

- (2.3.1) *Ze zamíchaných karet rozdáme dvěma hráčům po pěti kartách. Jsou výběry karet, které dostanou, nezávislé?*
- (2.3.2) *Hodíme dvěma stejnými kostkami. Je jev na obou padne stejné číslo nezávislý s jevem na první padne dvojka?*
- (2.3.3) *Jaká je pravděpodobnost, že při současném hodu třemi stejnými kostkami padnou čísla 1, 3 a 5?*
- (2.3.4) *Jaká je pravděpodobnost, že při pěti hodech mincí padne vždy hlava?*
- (2.3.5) *Hráč dostává ze zamíchaných 32 karet do ruky pět karet. Jaká je pravděpodobnost, že dvakrát po sobě dostane do ruky královský pokr?*

## 2.4 Střední hodnota

Dále se podívejme na náhodné procesy, jejichž výsledkem je číslo, obvykle přirozené. (Omezíme se jen na procesy s konečně mnoha možnými výsledky.)

**Definice:** Výsledek (předem neurčené číslo) takového náhodného procesu budeme nazývat *náhodnou proměnnou*.

**Definice:** Nechť náhodná proměnná  $X$  nabývá  $k$  možných hodnot z množiny  $X \in \{h_1, h_2, \dots, h_k\}$ , kde  $X_i$  nastává s pravděpodobností  $p_i$ , a  $p_1 + p_2 + \dots + p_k = 1$ . *Střední hodnotou proměnné  $X$*  je pak číslo

$$EX = p_1 \cdot h_1 + p_2 \cdot h_2 + \dots + p_k \cdot h_k.$$

**Komentář:** Zhruba řečeno, střední hodnota náhodné proměnné udává, jaký asi bude průměr získaných hodnot náhodné proměnné při mnoha opakováních náhodného procesu. Určit střední hodnotu má velký význam při statistických analýzách různých náhodných procesů, třeba známý algoritmus třídění **quicksort** může počítat i velmi dlouho, ale střední hodnota času jeho běhu je nejlepší ze všech běžných třídících algoritmů.

Praktický výpočet středních hodnot v situacích složených výběrů lze výrazně ulehčit pomocí následujících dvou tvrzení.

**Věta 2.6.** *Pro libovolné dvě náhodné proměnné  $X, Y$  platí*

$$E(X + Y) = EX + EY.$$

**Věta 2.7.** *Pro libovolné dvě **nezávislé** náhodné proměnné  $X, Y$  platí*

$$E(X \cdot Y) = EX \cdot EY.$$

**Komentář:** Pro příklady na střední hodnotu se podívejme na hody kostkami.

Jaká je střední hodnota (průměr) čísel padlých na šestistěnné kostce? Jednoduchým výpočtem vyjde  $EK = \frac{1}{6}(1 + 2 + 3 + 4 + 5 + 6) = \frac{21}{6} = 3.5$ .

Jaká je střední hodnota součtu čísel padlých na dvou šestistěnných kostkách? S využitím předchozího výsledku  $E(K_1 + K_2) = EK_1 + EK_2 = 3.5 + 3.5 = 7$ .

Jaká je střední hodnota součinu čísel padlých na dvou šestistěnných kostkách? S využitím předchozího výsledku  $E(K_1 \cdot K_2) = EK_1 \cdot EK_2 = 3.5 \cdot 3.5 = 12.25$ .

Vraťme se nyní k našemu intuitivnímu pojetí pravděpodobnosti jako očekávání četnosti opakování jevu. Pokud například hodíme  $n$ -krát mincí, pak pravděpodobnost  $\frac{1}{2}$  jedné strany mince znamená, že zhruba  $n/2$ -krát padne každá ze stran mince. Pokud je naše matematická formalizace pravděpodobnosti správná, měla by nám tentýž výsledek říci střední hodnota počtu hlav při  $n$  hodech mincí. Jak tomu skutečně je, ukazuje následující příklad.

**Příklad 2.8.** *Jaký je průměrný počet hlav padlých při  $n$  hodech mincí?*

Pokud hlavě mince přiřadíme hodnotu 1, máme tak  $n$  náhodných proměnných  $X_i \in \{0, 1\}$ ,  $i = 1, \dots, n$  příslušejících jednotlivým hodům mince. Celkový počet hlav je dán náhodnou proměnnou  $X = X_1 + \dots + X_n$ . Takže průměrný počet hlav je dle Věty 2.6

$$EX = E(X_1 + \dots + X_n) = EX_1 + \dots + EX_n = \frac{1}{2} + \dots + \frac{1}{2} = \frac{n}{2}.$$

To přesně odpovídá našemu vnímání pravděpodobnosti jako *relativní četnosti jevu*. □

**Příklad 2.9.** *Kolik je třeba průměrně hodů mincí, aby vyšly tři stejné výsledky?*

Je snadno vidět, že nejdříve tři stejné výsledky mohou nastat po třech hodech a nejpozději po pěti hodech (z dvou hlav a dvou orlů pět hodů nesložíme). S jakou pravděpodobností získáme stejné výsledky při třech hodech? Jsou možnosti buď tři hlav, nebo tři orlů, takže

$$p_3 = 2 \cdot \frac{1}{8} = \frac{1}{4}.$$

Až po pěti hodech získáme 3 stejné výsledky, pokud první čtyři hody budou rozděleny dva na dva (na posledním hodu pak již vlastně nezáleží). To se může stát v  $\binom{4}{2} = 6$  možnostech pro 4 hody, takže pravděpodobnost je

$$p_5 = 6 \cdot \frac{1}{16} = \frac{3}{8}.$$

Možnost, že 3 stejné výsledky získáme po čtyřech hodech je doplňková k předchozím dvěma a v součtu musí mít pravděpodobnost 1, proto

$$p_4 = 1 - p_3 - p_5 = 1 - \frac{1}{4} - \frac{3}{8} = \frac{3}{8}.$$

Průměrný počet potřebných hodů je dle definice střední hodnoty

$$N = p_3 \cdot 3 + p_4 \cdot 4 + p_5 \cdot 5 = \frac{3}{4} + \frac{3}{2} + \frac{15}{8} = \frac{33}{8} = 4.125.$$

(To je o trochu více než 4, což by jeden mohl intuitivně očekávat.)  $\square$

**Příklad 2.10.** (*varovný*) *Jaký je průměrný součin čísel horní a spodní stěny stejné kostky při hodech?*

Jak už víme, střední hodnota čísla na horní stěně je 3.5 a dolní stěně samozřejmě taky 3.5. Střední hodnota jejich součinu však *není*  $3.5 \cdot 3.5 = 12.25$ , protože tyto dva jevy **nejsou nezávislé**.

Místo toho střední hodnotu součinu spočítáme podle definice

$$\frac{1}{6}(1 \cdot 6 + 2 \cdot 5 + 3 \cdot 4 + 4 \cdot 3 + 5 \cdot 2 + 6 \cdot 1) = \frac{1}{6} \cdot 56 = 9 + \frac{1}{3}.$$

Proto si dávejme dobrý pozor na nezávislost jevů při násobení středních hodnot!  $\square$

### Úlohy k řešení

(2.4.1) *Jaká je střední hodnota počtu šestek padlých při hodu 10 šestistěnných kostek?*

(2.4.2) *Jaký je průměrný součet čísel horní a spodní stěny stejné kostky při hodech?*

(2.4.3) *Kolik je třeba průměrně hodů mincí, aby vyšly dva stejné výsledky?*

\* (2.4.4) *Kolik je třeba průměrně hodů mincí, aby padla první hlava?*

Návod: Uvědomte si, že teoreticky hlava nemusí padnout nikdy, ale pravděpodobnost samých orlů jde k nule.

## 2.5 Náhodné výběry

V závěrečném oddíle stručně shrneme nejčastěji se objevující typy *konečných uniformních náhodných výběrů* v diskrétní matematice:

**Náhodná podmnožina.** Z dané  $n$ -prvkové množiny vybíráme libovolnou z  $2^n$  jejích podmnožin, každou s pravděpodobností  $2^{-n}$ .

**Náhodná permutace.** Ze všech  $n!$  permutací dané  $n$ -prvkové množiny vybíráme libovolnou jednu s pravděpodobností  $1/n!$ .

**Náhodná kombinace.** Ze všech  $\binom{n}{k}$   $k$ -prvkových kombinací dané  $n$ -prvkové množiny vybíráme libovolnou jednu s pravděpodobností  $1/\binom{n}{k}$ .

**Náhodná posloupnost bitů.** Toto je poněkud jiný druh výběru – vybíráme libovolně dlouhou posloupnost z 0 a 1 tak, že každý další bit je vybírán s pravděpodobností  $\frac{1}{2}$  zcela nezávisle na všech předchozích bitech.

(Tj. každá vybraná podposloupnost této náhodné posloupnosti musí mít stejnou šanci se objevit.)

Poznámka: Kolem problému určit, *nakolik náhodná* je určitá posloupnost bitů, je vystavěna rozsáhlá matematická teorie, ale problém stále není vyřešený k plnému uspokojení (je to “velká věda”).

### Úlohy k řešení

(2.5.1) *Jakou pravděpodobnost má náhodná 7-prvková podmnožina 10-prvkové množiny?*

(2.5.2) *Jakou pravděpodobnost má náhodná permutace 4 prvků?*

(2.5.3) Jaká je pravděpodobnost, že náhodná podmnožina 5-prvkové množiny má jeden prvek?

(2.5.4) Jaká je pravděpodobnost, že náhodná posloupnost tří bitů vyjde '011'?

### Rozšiřující studium

Zájemce o hlubší studium pravděpodobnosti jako takové odkazujeme na klasické učebnice statistiky, kterými se zde nezabýváme. Pro rozšíření znalostí o specifické oblasti kombinatorické pravděpodobnosti doporučujeme [5, Kapitola 9], která se mimo jiné široce věnuje i použití tzv. pravděpodobnostních důkazů. Pokud čtenáře zajímají základy použití kombinatorické pravděpodobnosti v algoritmech, odkazujeme jej nejprve na [3].

## 2.6 Cvičení: Úlohy na pravděpodobnost

**Příklad 2.11.** Představme si, že hodíme dvěma poctivými kostkami – jednou 8-stěnnou a druhou 12-stěnnou. Jaká je pravděpodobnost, že na obou padne stejné číslo?

Na menší kostce může padnout vlastně cokoliv. Pak máme šanci  $\frac{1}{12}$ , že se hod větší kostky trefí zrovna do toho samého čísla. Výsledek tedy je

$$\frac{1}{12}.$$

□

**Příklad 2.12.** Jaká je pravděpodobnost, že první i poslední karta v náhodně zamíchaném balíčku klasických 32 karet budou srdce?

Jelikož nás vůbec nezajímá pořadí karet mimo první a poslední, počítáme s  $32 \cdot 31$  elementárními jevy určenými volbou první karty a následně volbou poslední karty různé od první. Z toho jev *obě budou srdce* má  $8 \cdot 7$  možností – volba prvního srdce je libovolná z osmi, druhé srdce musí být voleno ze zbylých sedmi srdcí. Výsledek tedy při uniformní pravděpodobnosti je

$$\frac{8 \cdot 7}{32 \cdot 31} = \frac{7}{124}.$$

□

**Příklad 2.13.** Hodíme najednou dvěma šestistěnnými kostkami. Jaká je pravděpodobnost, že menší z obou čísel bude 3?

Hody obou kostek jsou nezávislé, ale my si (jako v Příkladě 2.5) musíme dát pozor, na které kostce padne 3 jako menší z obou čísel. Nechť kostky jsou třeba červená a zelená. Úloha se nám rozpadá na tři disjunktí možnosti – na červené padne menší číslo, nebo na zelené padne menší, nebo na obou padne stejně. Vezměme třeba první možnost, kde pravděpodobnost 3 na červené kostce je  $\frac{1}{6}$  a pravděpodobnost čísla většího než 3 na zelené kostce je  $\frac{1}{2}$ . Tyto jevy jsou nezávislé. Obdobná úvaha platí pro další dvě možnosti, takže sečtením vyjde

$$P = \frac{1}{6} \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{6} + \frac{1}{6} \cdot \frac{1}{6} = \frac{7}{36}.$$

□

**Příklad 2.14.** Představme si, že házíme dvěma odlišenými šestistěnnými kostkami. Pro každý hod vynásobíme číslo první kostky součtem obou čísel. Jaká je střední hodnota výsledku?

Bohužel se nejedná o nezávislé jevy, takže nelze přímo aplikovat Větu 2.7. Označme  $K$  a  $K'$  náhodné veličiny udávající čísla padlá na našich kostkách. Úkolem tedy je spočítat střední hodnotu

$$E(K \cdot (K + K')) .$$

Dle definice můžeme rozepsat tuto střední hodnotu na celkem 36 sčítanců. My však lépe můžeme náhodnou veličinu roznásobit a použít Větu 2.6, která nevyžaduje nezávislost:

$$E(K \cdot (K + K')) = E(K^2 + K \cdot K') = E(K^2) + E(K \cdot K')$$

Nyní již dle Věty 2.7 vypočteme

$$E(K \cdot K') = E(K) \cdot E(K') = \left(\frac{7}{2}\right)^2$$

a dle definice spočteme

$$E(K^2) = \frac{1}{6} \cdot (1 + 4 + 9 + 16 + 25 + 36) = \frac{91}{6} .$$

Celkový výsledek je

$$E(K \cdot (K + K')) = \frac{49}{4} + \frac{91}{6} .$$

□

### Úlohy k řešení

- (2.6.1) Představme si, že hodíme najednou dvěma kostkami – jednou 10-stěnnou a druhou 6-stěnnou. Jaká je pravděpodobnost, že na obou padne stejné číslo?
- (2.6.2) Jaká je pravděpodobnost, že první i poslední karta v náhodně zamíchaném balíčku klasických 32 karet budou esa?
- (2.6.3) Jaká je pravděpodobnost, že první čtyři karty v náhodně zamíchaném balíčku klasických 32 karet budou esa?
- (2.6.4) Jaká je pravděpodobnost, že při 6 hodech mincí vyjdou výsledky hlava/orel přesně napůl?
- (2.6.5) Zamícháme náhodně 32 klasických karet (4 barvy po 8 kartách) a vytáhneme první 2 z nich. Jaká je pravděpodobnost, že to
- budou obě esa,
  - budou král a dáma (v tomto pořadí),
  - nebude žádné eso,
  - budou obě stejné hodnoty (různé barvy)?

Návod: Nemusíte uvažovat všech 32! permutací, stačí brát jen všechny možné výběry na první a druhou kartu.

- (2.6.6) Ve třídě je 17 studentů a 7 studentek. Z nich vybereme uniformně náhodně jednu neuspořádanou dvojici. Jaká je pravděpodobnost, že vybraná dvojice je
- stejného pohlaví,
  - různého pohlaví?

- (2.6.7) Jaká je pravděpodobnost, že po zamíchání balíčku 32 karet nebude mezi prvními pěti kartami žádné eso?

Návod: Přestože úloha mluví o zamíchání karet, uvědomte si, že není třeba se ohlížet na jejich pořadí.

- (2.6.8) Hodíme najednou dvěma šestistěnnými kostkami. Jaká je pravděpodobnost, že větší z obou čísel bude 3?

- \*(2.6.9) Jaká je střední hodnota počtu políček, o které se vaše figurka v kole přesune ve hře "Člověče, nezlob se", pokud se po třetí šestce za sebou již znovu nehází?

(Uvažujeme hody šestistěnnou kostkou, znovu házíme jen padne-li na kostce 6, až tři hody za sebou. Součet hodů určuje počet políček.)

- (2.6.10)** V šuplíku máme volně rozházených po 6 ponožkách od každé z barev žlutá, modrá, zelená. Kolik jednotlivých ponožek průměrně musíme poslepu vytáhnout, abychom dostali jednobarevný pár?

Formálně se jedná o výpočet střední hodnoty počtu vytažených ponožek. Ponožky jsou stejné až na barvu, není ani rozlišení mezi levou a pravou.

Návod: Nejdříve získáme jednobarevný pár po vytažení dvou ponožek, nejpozději po vytažení čtyř. Pak stačí spočítat pravděpodobnosti jednotlivých vytažení a udělat vážený průměr podle definice.

- (2.6.11)** Hodíme dvěma kostkami. Necht'  $A_k$  je jev "na první kostce padlo  $k$  a  $B_s$  je jev "na obou padl součet  $s$ ". Pro jaká  $k, s$  jsou jevy  $A_k, B_s$  nezávislé?

Návod: Fyzický postup výběru jevů  $A_k$  a  $B_s$  zajisté nezávislý není, neboť hod první kostky ovlivní i součet obou. Přesto za jistých podmínek lze nezávislost ověřit dle definice. Pokuste se o to.

- (2.6.12)** Jak jistě víte, každý magnet má dva opačné póly, které se přitahují. Barevné dětské magnetky mají na sobě umělohmotnou čepičku překrývající celou jejich stranu, takže jen druhá strana, tj. jeden z jejich pólů, je dostupná k přitáhnutí. Tyto magnetky jsou balené po 40 kusů, 10 od každé ze čtyř barev. Předpokládejme, že nezakryté póly těchto magnetků jsou zvoleny náhodně s pravděpodobností  $\frac{1}{2}$ . Jaká je pak pravděpodobnost, že těchto 40 magnetků lze pospojovat do  $5 \times 4$  stejnobarevných dvojic (že každá dvojice se navzájem přitahuje opačnými nezakrytými póly)?

Návod: Kolik asi musí být magnetků od každého nezakrytého pólu v každé barvě?

### 3 Důkazy v diskrétní matematice

#### Úvod

Jednou z hlavních deviz matematiky je její exaktnost, tedy její schopnost jednoznačně (mimo jakoukoliv pochybnost) dokazovat svá tvrzení. Koncept matematického důkazu má za sebou dlouhý historický vývoj již od dob Euklida a jeho axiomů geometrie, přes Bolzanovy práce formalizující moderní koncept důkazu, až po bouřlivý rozvoj matematické logiky v počátcích 20. století po objevení Russelova paradoxu.

V moderním pohledu je matematický důkaz nahlížen jako posloupnost ověřitelných elementárních kroků vedoucích od známých a předpokládaných faktů k požadovanému tvrzení. Když se nad tímto konceptem hlouběji zamyslíme, vidíme, že pokud se chceme někam dostat, nemůžeme začínat "z ničeho". Proto, jak si uvědomil v dávných dobách již zmíněný Euklides, každá matematická teorie potřebuje své základní předpoklady, zvané "axiomy". V diskrétní matematice jsou tyto základní předpoklady tvořeny axiomami tzv. Peanovy aritmetiky, což je vlastně vznešený název pro dobře známá základní fakta o přirozených číslech, doplněná principem matematické indukce.

Schopnost podat exaktní matematický důkaz svých tvrzení je však důležitá i mimo oblast matematiky samotné. Například při návrhu kritických informačních systémů, na jejichž správné funkci závisí lidské životy, je více než žádoucí podat skutečný důkaz jejich správné funkce ve všech případech, ne jen v testovacích simulacích. Důkazové metody diskrétní matematiky, obzvláště princip matematické indukce, jsou v těchto situacích vhodné a dobře použitelné.

#### Cíle

Obecným cílem je ukázat čtenáři principy, na kterých staví exaktní matematické důkazy, a naučit je předkládané důkazy správně chápat. Tyto znalosti a schopnosti budou (mlčky) předpokládány pro celý zbytek studijního textu. (To však neznamená, že by se studenti měli učit uvedené důkazy nazpaměť, byla by to naprosto zbytečná ztráta času! Jde o pochopení principů.) U studentů, kteří aspirují na nejlepší známku, očekáváme i schopnost formulovat vlastní důkazy běžných (a nepříliš složitých) tvrzení.

### 3.1 Základní logické symboly

Pro začátek si zopakujeme základní logické pojmy a symboly, které byste již měli znát z předchozího studia. Úvodní opakování z matematické logiky...

- Logické hodnoty jsou 1/0, neboli *True/False* (Pravda/Nepřavda).
- Logické *spojky* jsou  $\neg X$  (NOT),  $X \wedge Y$  (AND),  $X \vee Y$  (OR).
- *Implikace*  $X \Rightarrow Y$  znamená “pokud  $X$  je pravda, musí  $Y$  být pravda”.  
*Ekvivalence*  $X \iff Y$  znamená “ $X$  je pravda zároveň s  $Y$  pravdivým”.
- *Kvantifikace*:
  - $\forall x : P(x)$  znamená, že  $P(x)$  je pravdivé pro všechny volby proměnné  $x$  (z dohodnuté či vyznačené množiny “ $\forall x \in M$ ”).
  - $\exists x : P(x)$  znamená, že  $P(x)$  je pravdivé pro aspoň jednu volbu proměnné  $x$  (z dohodnuté či vyznačené množiny “ $\exists x \in M$ ”).

**Komentář:** Pokusme se například vyjádřit známou “počítačovou” logickou operaci XOR pomocí výše uvedených operací matematické logiky.

Jak známo,  $a \text{ XOR } b$  je 1 (True) právě když  $a, b$  mají různé logické hodnoty, což je přesně negací ekvivalence, tj.

$$a \text{ XOR } b \equiv \neg(a \iff b).$$

### 3.2 Pojem matematického důkazu

Než uvedeme, co je matematický důkaz, musíme si nejprve ujasnit, co bychom měli dokazovat, tj. co je matematické tvrzení.

**Definice:** Matematické *tvrzení* (výrok) se skládá z přesně formulovaných *předpokladů* a ze *závěru*.

Záměrně zde uvádíme jen velmi neformální opis nutných pojmů, neboť našim cílem je ukázat čtenáři základní principy a naučit jej chápat běžné jednoduché důkazy. Uvedení přesných logických definic by problematiku pro běžného čtenáře jen více zamžilo.

**Definice:** Slovně opsáno, *matematický důkaz* nějakého tvrzení  $P$  je konečná posloupnost kroků, kde každý krok obsahuje výrok, který je jedním z

- *axiomů* – tj. obecně platných či předpokládaných faktů,
- *předpokladů* tvrzení  $P$ ,
- *výroků odvozených* z předchozích kroků za použití některého z platných odvozovacích pravidel,

a kde poslední krok obsahuje jako výrok *závěr tvrzení*  $P$ .

Přitom volba systému axiomů závisí na matematické teorii, ve které pracujeme, a je tudíž různá v různých oblastech matematiky. Výběr použitelných odvozovacích pravidel je vlastností použité logiky a bude obvykle obsahovat jen přirozená odvození, nad kterými se v praxi ani nerozmýšlíme.

**Poznámka:** Pamatujme, že různé matematické teorie mohou pracovat s různými množinami axiomů (dokonce i s různými odvozovacími pravidly), a tudíž dokazovat jiná tvrzení. Pro příklad můžeme zajít dokonce až do dávné historie – k axiomům *Euklidovské geometrie*, konkrétně k axiomu o rovnoběžkách. Po dlouhá dvě tisíciletí se matematici snažili dokázat platnost axiomu o rovnoběžkách z ostatních Euklidových axiomů, až teprve před zhruba dvěma stoletími se přišlo na to, že změnou axiomu o rovnoběžkách vzniknou úplně jiné, tzv. neeuklidovské, geometrie.



Pro ilustraci konceptu matematického důkazu jsme zvolili dobře známý fakt z algebry.

**Příklad 3.1.** Dokažme v libovolném algebraickém (polo)okruhu pravidlo “agresivity nuly při násobení”, tj. že  $x \cdot 0 = 0$  pro všechna  $x$ .

Algebraický okruh je početní struktura s operacemi  $+$ ,  $\cdot$ , která splňuje několik základních a přirozených aritmetických vlastností – axiomů. Z nich zde využijeme následující axiomy:

$$\forall a : a + 0 = 0 + a = a \quad (1)$$

$$\forall a, b, c : a + c = b + c \Rightarrow a = b \quad (2)$$

$$\forall a, b, c : a \cdot (b + c) = a \cdot b + a \cdot c \quad (3)$$

Dobře si nyní promysleme, co je vlastně cílem našeho snažení. Někdo by se mohl ptát, proč dokazovat pravidlo  $x \cdot 0 = 0$ , když to je přece jasné a každý to ví ze školy. Toto pravidlo však není zahrnuto mezi axiomy a důkaz potřebuje. A především, výsledkem našeho snažení bude důkaz, že fakt  $x \cdot 0 = 0$  platí ve všech možných početních strukturách, které splňují výše uvedené axiomy; přitom se vůbec nemusí jednat o počítání s čísly, jak je známe ze školy, ale o úplně jiné objekty, třeba geometrické.

Důkaz výroku  $x \cdot 0 = 0$  v krocích:

- Dosazením  $a = 0$  do (1) odvodíme  $0 + 0 = 0$ .
- Vynásobením předchozího číslem  $x$  získáme  $x \cdot (0 + 0) = x \cdot 0$ .
- Dosazením  $a = x$ ,  $b = c = 0$  do (3) odvodíme  $x \cdot (0 + 0) = x \cdot 0 + x \cdot 0$ .
- Spojením předchozích vztahů (tranzitivita  $=$ ) vyjde  $x \cdot 0 = x \cdot 0 + x \cdot 0$ .
- Dosazením  $a = x \cdot 0$  do (1) dále vyjde  $x \cdot 0 = 0 + x \cdot 0 = x \cdot 0 + x \cdot 0$ .
- Z (2) pro  $c = x \cdot 0$  nakonec dostaneme  $0 = x \cdot 0$ .

(Pochopitelně v praxi důkaz zapisujeme “zkratkou”, ale naše zkratky musí být korektní, tj. rozepsatelné do takovýchto elementárních kroků.)  $\square$

### 3.3 Princip matematické indukce

Princip matematické indukce je jedním z velmi důležitých axiomů přirozených čísel. Místo jeho formálního zavedení jako axiomu si jej pro lepší pochopitelnost uvedeme ve formě *odvozací metody matematických důkazů*:

#### Metoda 3.2. *Matematická indukce*

Mějme tvrzení  $P(n)$  s celočíselným parametrem  $n$ . Nechť následující platí:

- Tvrzení  $P(n_0)$  je pravdivé, kde  $n_0 = 0$  či 1, nebo i pro obecné celé  $n_0$ . (Nazýváno *základ indukce*.)
- Pro libovolné přirozené  $n$  z platnosti tvrzení  $P(n)$  plyne platnost tvrzení  $P(n + 1)$ . (Tvrzení  $P(n)$  je v tomto kroku nazýváno *indukční předpoklad*.)

Pak  $P(n)$  platí pro všechna přirozená  $n \geq n_0$ .

Jak tedy můžeme princip matematické indukce použít? Typicky je použijeme v situaci, kdy máme dokázat platnost nějakého tvrzení s celočíselným parametrem, třeba  $n$ . Přitom platnost tohoto tvrzení by měla být poměrně zřejmá pro malá  $n$  a zvýšení parametru  $n$  o 1 by nemělo příliš změnit situaci.

**Komentář:** Například, představme si, že dokazujeme korektnost nějakého algoritmu – vidíme, že po spuštění algoritmus je v konzistentním stavu, a po každém dalším kroku se konzistence stavu neporuší. Pak algoritmus zůstane v konzistentním stavu po celou dobu svého běhu podle principu matematické indukce. Nebo se podívejme na následující důležitý ilustrační příklad.

**Věta 3.3.** Každá  $n$ -prvková množina má právě  $2^n$  podmnožin (včetně prázdné). Formálně

$$|2^X| = 2^{|X|}.$$

**Důkaz** matematickou indukcí podle  $n$ :

- Pro  $n = 0$  tvrzení platí, neboť prázdná množina má jedinou podmnožinu, opět prázdnou.
- Nechť nyní tvrzení platí pro  $n \geq 0$ . Vezmeme libovolnou množinu  $X$  o  $n + 1 > 0$  prvcích. Zvolme prvek  $a \in X$  a označme  $X' = X \setminus \{a\}$ ,  $|X'| = n$ . Potom všech podmnožin  $P' \subseteq X'$  je podle indukčního předpokladu  $2^n$ . Pro prvek  $a$  navíc máme nezávislý výběr ze dvou možností: buď  $a$  dáme do podmnožiny  $P'$ , nebo nedáme. Celkem je tak  $2 \cdot 2^n = 2^{n+1}$  možností volby podmnožiny  $P \subseteq X$ , cbd.

Důkaz je hotov podle principu matematické indukce.  $\square$

Obdobně lze dokázat následující tvrzení:

**Věta 3.4.** Je právě  $a^b$  všech zobrazení z libovolné  $b$ -prvkové množiny do  $a$ -prvkové množiny. Formálně

$$|A^B| = |A|^{|B|}.$$

**Poznámka:** Zobrazení z  $b$ -prvkové množiny do  $a$ -prvkové množiny se také jinak nazývají  *$b$ -prvkové variace s opakováním z  $a$  prvků*. My však dáváme přednost tomu je nazývat zobrazeními, neboli je vnímat jako posloupnost  $b$  nezávislých jednoprvkových výběrů z  $a$ -prvkové množiny.

### 3.4 Důkazy vzorců pro výběry

Nyní si formálně dokážeme vzorce pro výběry prvků uvedené v Lekci 1. Použijeme přitom matematickou indukcí a také Metodu 1.5 dvojího počítání.

**Věta 3.5.** Počet všech permutací  $n$ -prvkové množiny je  $n!$ , pro každé  $n \geq 0$ .

**Důkaz** indukcí podle  $n$ : Tvrzení platí pro  $n = 0$ , protože žádné prvky lze uspořádat jen jedním způsobem (stejně tak jeden prvek).

Mějme nyní  $n \geq 0$  a množinu  $P$  o  $n + 1$  prvcích, předpokládejme pro jednoduchost  $P = \{1, 2, \dots, n + 1\}$ . Zvolme první prvek  $p \in P$  naší permutace jedním z  $n + 1$  způsobů. Chtělo by se přímo říct, že potom už je volba zbytku permutace  $P \setminus \{p\}$  nezávislá na volbě prvního  $p$ , ale to formálně není pravda. Aby tomu tak skutečně bylo, musíme si zbylé prvky  $P \setminus \{p\}$  nejprve “přečíslovat” na  $\{1, 2, \dots, n\}$ , což počet voleb neovlivní.

Pak už však zbytek plyne jasně –  $n$ -prvková množina  $\{1, 2, \dots, n\}$  má podle indukčního předpokladu  $n!$  permutací, proto  $P$  má celkem  $(n + 1) \cdot n! = (n + 1)!$  permutací. To je přesně to, co chceme.  $\square$

**Věta 3.6.** Počet všech  $k$ -prvkových variací z  $n$ -prvkové množiny je  $\frac{n!}{(n-k)!}$ , pro každé  $n \geq k \geq 0$ .

**Důkaz** metodou dvojího počítání:

Budeme se dvěma způsoby dívat na výběr permutací  $n$ -prvkové množiny. Jak již víme, lze tyto permutace vybrat  $n!$  různými způsoby. Na druhou stranu lze vzít některou  $k$ -prvkovou variaci, její prvky dát na začátek permutace v jejich pořadí a zbylých  $n - k$  prvků seřadit za nimi jedním z  $(n - k)!$  různých způsobů. Z různých variací tímto

postupem zřejmě získáme různé výsledné permutace, a přitom každou permutaci lze získat z variace vybírající jejích prvních  $k$  prvků.

Označíme-li  $x$  neznámý počet všech  $k$ -prvkových variací z  $n$ -prvkové množiny, lze výše popsaným postupem vytvořit právě  $x \cdot (n - k)!$  všech různých permutací  $n$ -prvkové množiny. Proto platí

$$x \cdot (n - k)! = n!,$$

$$x = \frac{n!}{(n - k)!}.$$

□

**Věta 3.7.** *Počet všech  $k$ -prvkových kombinací z  $n$ -prvkové množiny je  $\binom{n}{k}$ , pro každé  $n \geq k \geq 0$ .*

**Důkaz** metodou dvojího počítání:

Nyní budeme dvojím způsobem počítat všechny  $k$ -prvkové variace z  $n$ -prvkové množiny. Na jednu stranu už víme, že jich je  $\frac{n!}{(n-k)!}$ , na druhou stranu můžeme z *jedné*  $k$ -prvkové kombinace vygenerovat celkem  $k!$  různých variací uspořádáním prvků této kombinace. Označíme-li tedy  $x$  neznámý počet všech  $k$ -prvkových kombinací z  $n$ -prvkové množiny, dostaneme obdobně jako v předchozím důkaze

$$x \cdot k! = \frac{n!}{(n - k)!},$$

$$x = \frac{n!}{k! \cdot (n - k)!} = \binom{n}{k}.$$

□

### 3.5 Vztahy s kombinačními čísly

Kombinační čísla splňují množství zajímavých vztahů a identit. (Zabývá se jimi dokonce celá samostatná část diskrétní matematiky.) Tři základní z nich uvádíme zde. Každá z nich má přirozenou oporu v pohledu na počty výběrů kombinací.

**Fakt.** Pro všechna  $n \geq 0$  platí

$$\binom{n}{0} = \binom{n}{n} = 1.$$

**Fakt.** Pro všechna  $n \geq k \geq 0$  platí

$$\binom{n}{k} = \binom{n}{n - k}.$$

**Lema 3.8.** *Pro všechna  $n \geq k \geq 0$  platí*

$$\binom{n}{k} + \binom{n}{k + 1} = \binom{n + 1}{k + 1}.$$

**Důkaz:**

$$\begin{aligned} \binom{n}{k} + \binom{n}{k + 1} &= \frac{n!}{k! \cdot (n - k)!} + \frac{n!}{(k + 1)! \cdot (n - k - 1)!} = \\ &= \frac{n! \cdot (k + 1) + n! \cdot (n - k)}{(k + 1)! \cdot (n - k)!} = \frac{n! \cdot (n + 1)}{(k + 1)! \cdot (n - k)!} = \binom{n + 1}{k + 1}. \end{aligned}$$

□

Tyto tři uvedené fakty mohou také posloužit jako *definice kombinačních čísel*. (Viz. Pascalův trojúhelník.)



## Úlohy k řešení

(3.5.1) Upravte následující výraz na jediné kombinační číslo:

$$\binom{2n}{n-1} + \binom{2n}{n+2}$$

\*(3.5.2) Upravte následující výraz na jediné kombinační číslo:

$$\binom{n}{0} + \binom{n+1}{1} + \binom{n+2}{2} + \binom{n+3}{3}$$

## 3.6 Důkazy “počítáním”

Někdy jsme při důkazech tvrzení postaveni do situace, že máme ukázat na existenci nějakého objektu nebo vlastnosti, které nejsme schopni konkrétně předvést nebo specifikovat. V takové situaci se hodí tzv. nekonstruktivní metody důkazu, které jistým způsobem “spočítají” existenci řešení, aniž řešení přímo specifikují. Nejznámějším příkladem je následující princip, který použijeme pro ilustraci:

### Metoda 3.11. *Dirichletův princip*

*Rozmístíme-li  $\ell + 1$  (nebo více) objektů do  $\ell$  přihrádek, v některé přihrádce musí být aspoň dva objekty.*

**Příklad 3.12.** *Mezi čtyřmi přirozenými čísly vždy najdeme dvě, jejichž rozdíl je dělitelný číslem 3.*

Nechť přihrádkami jsou zbytkové třídy při dělení číslem 3. Máme tedy pro naše 4 čísla tři přihrádky značené 0, 1, 2. Podle Dirichletova principu do některé z přihrádek padnou dvě z čísel  $x, y$ , ale pak  $x - y$  dává zbytek 0 při dělení 3, tudíž jsme našli požadovanou dvojici.  $\square$

Na závěr lekce si ukážeme na příkladech ještě jeden zajímavý postup dokazování tvrzení v diskrétní matematice *počítáním možností*. Zhruba řečeno, dokazujeme existenci požadované možnosti tím, že spočteme jiné možnosti a ukážeme, že je jich méně než všech možností.

**Komentář:** Pro ilustraci: Do sledovaného objektu vešly včera tři osoby a dnes jej jen dvě osoby opustily. To znamená, že v objektu aspoň jedna osoba zůstává, přestože ji nikde nevidíme.

**Příklad 3.13.** *Na letním táboře 29 dětí stráví celkem 16 dní a 15 nocí. Každou noc jsou dva z táborníků na hlídce. Dokažte, že některé z dětí musí jít na hlídku za celý tábor aspoň dvakrát.*

To je velice snadné počítání: Je-li celkem třeba 15 nočních hlídek, *potřebujeme*  $15 \cdot 2 = 30$  *táborníků* na jejich pokrytí, pokud se žádný nemá opakovat. To však tak nejde, když je na táboře pouze  $29 < 30$  dětí.  $\square$

**Příklad 3.14.** *8 kamarádů jelo na 9 dní dovolené. Každý den některá (jedna) trojice z nich šla na výlet. Dokažte, že někteří dva z nich ani jednou nebyli spolu na výletě.*

Rozebírání možností by asi k ničemu nevedlo. . .

Důkaz počítáním je však opět snadný: Jedna trojice má celkem 3 dvojice, proto po 9 dnech se mohlo vystřídat *nejvýše*  $9 \cdot 3$  dvojic ve *výletních trojicích*, ale  $9 \cdot 3 = 27 < \binom{8}{2} = 28$ , jedna dvojice nám zde schází.  $\square$

## Úlohy k řešení

(3.6.1) Existují na VŠB dva studenti se stejným posledním čtyřčíslným rodného čísla? (A dokážete je najít?)

(3.6.2) Kdysi v dávných dobách se v MHD štípaly lístky, třeba vyražením čtveřice čísel z 12. Mohli jste se spolehnout, že pokud jste si schovávali své lístky, tak nejpozději po 500 jízdách jste mohli některý lístek použít znovu?

## Rozšiřující studium

Princip důkazu matematickou indukcí je podrobně vysvětlen v [5, Oddíl 1.3] a také v jiných učebnicích, třeba algebry. Důkazy mnoha početních kombinatorických vztahů lze najít v [5, Kapitola 2 a Kapitola 10], jedná se však o dosti obtížnou látku.

## 3.7 Cvičení: Příklady matematických důkazů

**Příklad 3.15.** Dokažme indukcí podle  $k$ , že číslo  $(2^{6k} - 1)$  je dělitelné sedmi.

Budeme postupovat indukcí podle parametru  $k \geq 0$ .

- Pro  $k = 0$  platí, že  $2^0 - 1 = 0$  je dělitelné sedmi.
- Nechť nyní tvrzení platí pro nějaké  $k$ . Podívejme se, o kolik se změní hodnota výrazu  $(2^{6k} - 1)$  při zvětšení  $k$  o jedna:

$$(2^{6(k+1)} - 1) - (2^{6k} - 1) = 2^6 \cdot 2^{6k} - 2^{6k} = 2^{6k} \cdot (2^6 - 1) = 2^{6k} \cdot 63$$

Důležitým faktem nyní je, že číslo 63 je dělitelné sedmi. To ale znamená, že pokud  $(2^{6k} - 1)$  bylo dělitelné sedmi podle indukčního předpokladu, bude sedmi dělitelné i  $2^{6(k+1)} - 1$ .

Formálně tento indukční krok zapíšeme následovně: Nechť dle indukčního předpokladu  $2^{6k} - 1 = 7\ell$ , kde  $\ell$  je nějaké přirozené číslo. Pak

$$\begin{aligned} (2^{6(k+1)} - 1) &= 2^6 \cdot 2^{6k} - 1 = (2^6 - 1) \cdot 2^{6k} + 2^{6k} - 1 = 63 \cdot 2^{6k} + 7\ell = \\ &= 7 \cdot (9 \cdot 2^{6k} + \ell), \end{aligned}$$

takže i tento výraz je dělitelný sedmi. □

**Příklad 3.16.** Dokažme následující identitu s kombinačními čísly:

$$\binom{n}{0}^2 + \binom{n}{1}^2 + \binom{n}{2}^2 + \dots + \binom{n}{n-1}^2 + \binom{n}{n}^2 = \binom{2n}{n}$$

Představme si  $2n$ -prvkovou množinu  $A = [1, 2n]$ , rozloženou na dvě disjunktní podmnožiny  $B = [1, n]$ ,  $C = [n+1, 2n]$ ,  $B \cup C = A$ . Budeme postupovat metodou dvojího počítání  $n$ -prvkových kombinací nad  $A$ :

- Na jednu stranu je těchto kombinací  $\binom{2n}{n}$ .
- Na druhou stranu si představme, že vybereme  $k$ -prvkovou podmnožinu  $B' \subseteq B$  a  $k$ -prvkovou podmnožinu  $C' \subseteq C$ , pro libovolné  $k \in [0, n]$ . Vytvoříme pak množinu  $K = B' \cup (C \setminus C')$ . To lze podle principu nezávislých výběrů udělat právě  $\binom{n}{k}^2$  způsoby.

- A kolik má množina  $K$  prvků? Bez ohledu na  $k$  to je  $k + (n - k) = n$ . Takže v předchozím bodě jsme vlastně jiným způsobem vybírali  $n$ -prvkovou kombinaci nad  $A$ . Jelikož dvakrát vybíráme totéž, musí platit rovnost v počtu výběrů, takže sečtením přes všechny hodnoty volného parametru  $k$  dostaneme požadované

$$\sum_{k=0}^n \binom{n}{k}^2 = \binom{2n}{n}.$$

□

**Příklad 3.17.** Mějme 28 mincí stejné hodnoty, ze kterých je jedna falešná – lehčí než ostatní, a přesné rovnoramenné váhy bez závaží. Najděte postup, jak falešnou minci určit pouze použitím 4 vážení. Dokažte, že tři vážení na určení falešné mince nestačí.

- Rozdělme mince na tři hromádky po 9-9-10 a první dvě z hromádek zvažme proti sobě. Pokud je hromádka  $A$  těžší než  $B$ , je falešná mince mezi  $B$ . Při opačném výsledku je falešná mince mezi  $A$ . A při rovnováze musí být falešná mince ve třetí hromádce  $C$ .
- Totéž zopakujeme pro vybranou hromádku rozdělenou na 3-3-3 či 3-3-4 mince.
- Totéž zopakujeme pro vybranou hromádku rozdělenou na 1-1-1 či 1-1-2 mince.
- Případně ještě rozhodneme mezi posledními dvěma mincema.

Naopak chceme ukázat, že 3 vážení nestačí. Využijeme důkaz počítáním. Každé jedno vážení má jen tři možné výsledky, proto po třech váženích jsme schopni rozlišit pouze  $3^3 = 27$  různých možností, ale celkem je  $28 > 27$  možných voleb falešné mince. Proto, ať děláme co děláme, při dvou různých volbách falešné mince musíme odpovědět stejně, takže jednou špatně. □

### Úlohy k řešení

(3.7.1) Dokažte, že pro každé přirozené  $n$  je číslo  $n^3 - n$  dělitelné šesti.

(3.7.2) Od jaké hodnoty  $n$  platí nerovnost  $\binom{n}{2} < \binom{n}{3}$ ? Dokažte, že pak už uvedená nerovnost je stále platná.

(3.7.3) Dokažte vztah:

$$\binom{r}{r} + \binom{r+1}{r} + \binom{r+2}{r} + \dots + \binom{n-1}{r} + \binom{n}{r} = \binom{n+1}{r+1}$$

\*(3.7.4) Dokažte nerovnosti:

$$n^{n/2} \leq n! \leq \left(\frac{n+1}{2}\right)^n$$

\*(3.7.5) Dokažte platnost následujícího vztahu pro všechna přirozená  $n \geq 1$ :

$$\sum_{i=1}^n (2i-1) \cdot 3^i = (n-1) \cdot 3^{n+1} + 3$$

Návod: Nejsnadnější je důkaz matematickou indukcí. Podívejte se, jaký člen přibude vlevo, pokud se  $n$  zvýší na  $n+1$ , a o kolik se pak zvětší pravá strana.

## 4 Relace a zobrazení

### Úvod

Pro hlubší studium diskrétní matematiky je dále nezbytná jistá úroveň formalizace pojmů a vztahů. Dostáváme se totiž do modernějších oblastí oboru, kde již nevystačíme jen s jednoduchými kombinatorickými představami o výběrech či seřazeních prvků, ale potřebujeme přesné vyjádření různých vztahů mezi objekty. K tomu si definujeme pojem relace a jejich speciálních případů uspořádání a ekvivalence.

Dále se budeme věnovat formalizaci pojmu zobrazení a jejich skládání. Důležitým druhem zobrazení pro nás je tzv. bijekce jedné množiny na druhou – zobrazení jednoznačně přiřazující prvky první množiny na prvky druhé množiny a naopak. Speciálně si tak ukážeme jiný možný pohled na permutace jako na bijekci množiny na sebe. Závěrem ještě uvedeme princip inkluze a exkluze užívaný pro určení počtu prvků ve sjednocení množin, které nejsou disjunktní.

### Cíle

Cílem této lekce je vysvětlit studujícímu formální pojmy relace, uspořádání a zobrazení a jejich použití k popisu různých matematických vztahů. Presentovaný formalismus je nezbytný pro hlubší studium diskrétní matematiky. Mimo to je zde popsán princip inkluze a exkluze mezi množinami.

### 4.1 Pojem relace

Pro připomenutí: *Kartézský součin* dvou množin je množinou všech uspořádaných dvojic vybraných po složkách z těchto množin, tj.

$$A \times B = \{(a, b) : a \in A, b \in B\}.$$

**Definice 4.1.** *Relace*  $R$  na množině  $A$  je libovolná podmnožina kartézského součinu  $A \times A = A^2$ , tj.

$$R \subseteq A^2.$$

Takto definovaná relace se také rozšířeně nazývá *binární relace*. Píšeme buď  $(x, y) \in R$ , nebo zkráceně  $xRy$  (třeba  $x = y$ ,  $x < y$ ).

**Komentář:** Jak již název napovídá, relace se využívají k popisu vztahů mezi prvky množiny (objekty). Dobře si uvědomme, že dvojice prvků buď v relaci je nebo není, třetí možnost není přípustná. Binární relace mluví o uspořádaných dvojicích prvků. V obecnosti se však může relace týkat i  $n$ -tic prvků, jak ukazuje další definice.

**Definice 4.2.**  *$n$ -ární relace*  $S$  na množině  $A$  je libovolná podmnožina kartézské mocniny  $A^n$ , tj.

$$S \subseteq A^n.$$

**Poznámka:** Relace dokonce může být i mezi různými množinami, tj. jako  $T \subseteq A \times B$ . Tomu se však zde nebudeme blíže věnovat.

**Definice 4.3.** (Binární) relace  $R$  na množině  $A$  je

- *reflexivní* pokud  $(x, x) \in R$  pro všechna  $x \in A$ ,
- *symetrická* pokud  $(x, y) \in R \Leftrightarrow (y, x) \in R$  pro všechna  $x, y \in A$ ,
- *antisymetrická* pokud z  $(x, y), (y, x) \in R$  vyplývá  $x = y$  pro všechna  $x, y \in A$ ,
- *tranzitivní* pokud z  $(x, y), (y, z) \in R$  vyplývá  $(x, z) \in R$  pro všechna  $x, y, z \in A$ .



Komentář: *Praktické příklady relací:*

- Relace rovnosti = je reflexivní, symetrická, tranzitivní i antisymetrická.  
Relace menší < je antisymetrická a tranzitivní,  $\leq$  je navíc reflexivní.  
Relace | dělitelnosti je také reflexivní, antisymetrická a tranzitivní.
- Pro příklad “manželství” je typicky symetrická relace.  
Vztah “má rád” je obvykle reflexivní, ale často není symetrický ani tranzitivní.  
Vlastnost “silnější než” je antisymetrická a obvykle i tranzitivní.
- Příkladem vícečetných ( $n$ -árních) relací je třeba výčet všech sešraných 11-členných výběrů mužstva na fotbal, které trenér může z dostupných hráčů sestavit.

## Úlohy k řešení

(4.1.1) Kolik uspořádaných dvojic nad množinou  $\{1, 2, 3, 4, 5\}$  patří do relace rovnosti =?

(4.1.2) Kolik uspořádaných dvojic nad množinou  $\{1, 2, 3, 4, 5\}$  patří do relace menší <?

(4.1.3) Kolik uspořádaných dvojic nad množinou  $\{1, 2, 3, 4, 5, 6\}$  patří do relace dělitelnosti?

\*(4.1.4) Mějme relaci soudělnosti nad přirozenými čísly, kde dvě čísla  $x, y$  jsou v relaci pokud mají společného dělitele většího než 1. Které z vlastností z Definice 4.3 tato relace má?

## 4.2 Uspořádání a ekvivalence

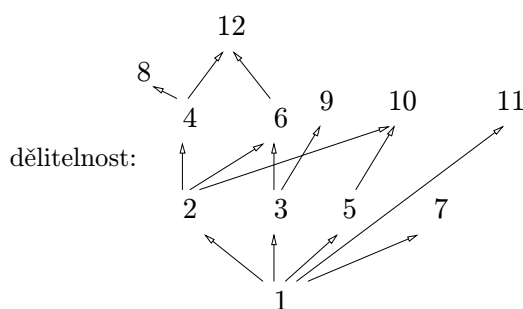
To jsou dva typy relací, se kterými se nejčastěji setkáme. . .

**Definice 4.4.** *Částečné uspořádání*  $\preceq$

je reflexivní, antisymetrická a tranzitivní binární relace na množině.

Pro vysvětlení, slovo “částečné” v definici znamená, že ne každé dva prvky lze vždy v částečném uspořádání porovnat. To je ostatně vidět třeba při relaci dělitelnosti.

Dále si všimněme podmínky reflexivity v definici, tj. že prvek je sám se sebou v relaci. V praxi se této vlastnosti uspořádání říká *neostrá nerovnost* a značí se vodorovnou čárkou pod znaménkem nerovnosti  $\preceq$ . Pochopitelně z neostré nerovnosti snadno získáme ostrou vyloučením rovnosti a naopak také.



Obrázek 4.1: *Hasseův diagram* – standardní způsob zobrazení částečného uspořádání (zde relace dělitelnosti) diagramem, kde “větší” prvky jsou kresleny výše než ty “menší” a kde od menších je kreslena šipka k “bezprostředně větším” prvkům (tj. šipky, které nevyplývají z tranzitivity relace).

Komentář: *Příklady částečných uspořádání:*

- Relace inkluze  $\subseteq$  (podmnožiny) mezi množinami. Dvě množiny mohou snadno být neporovnatelné inkluzí, jako třeba  $\{1, 2\}$  a  $\{1, 3, 4\}$ .

- Relace dělitelnosti  $|$  na přirozených číslech (viz také Obrázek 4.1).
- Srovnávání “lepšíh” a “horších” počítačů – ne každou dvojici lze rozhodnout. (Klíčovými kritérii jsou třeba cena, rychlost procesoru, velikost paměti, rychlost disku, atd.)

K částečným uspořádáním se dále vztahují následující pojmy.

**Definice:** Říkáme, že prvky  $a, b$  v částečném uspořádání  $\preceq$  jsou *neporovnatelné*, pokud neplatí ani  $a \preceq b$ , ani  $b \preceq a$ .

Říkáme, že posloupnost  $a_1, a_2, \dots, a_n$  tvoří *řetězec* v částečném uspořádání  $\preceq$ , pokud  $a_1 \preceq a_2 \preceq \dots \preceq a_n$ .

Prvek  $m$  je *minimální* v částečném uspořádání  $\preceq$  množiny  $A$ , pokud žádný jiný prvek  $A$  není menší než  $m$ , tj.  $\forall x \in A : x \preceq m \Rightarrow x = m$ .

Prvek  $m$  je *nejmenší* v částečném uspořádání  $\preceq$  množiny  $A$ , pokud každý jiný prvek  $A$  je větší než  $m$ , tj.  $\forall x \in A : m \preceq x$ .

*Maximální* a *největší* prvek v částečném uspořádání jsou definovány analogicky k minimálnímu a nejmenšímu prvku s obrácenými nerovnostmi.

**Komentář:** *Pro ilustraci:*

- 0 je nejmenší přirozené číslo dle velikosti.
- Mezi čísla 3, 4, 5, 6, 7, 8, 9 uspořádanými dělitelností není žádné nejmenší, ale třeba 3 nebo 7 jsou minimální prvky vzhledem k dělitelnosti. Na druhou stranu 6 ani 8 nejsou minimální prvky, protože 3 resp. 4 je dělí (jsou “menší”).
- Přirozená čísla nemají žádný největší ani maximální prvek dle velikosti. Kladná racionální čísla nemají ani žádný nejmenší ani minimální prvek.

**Definice:** Částečné uspořádání  $\preceq$  je *lineární* na množině  $A$ , pokud žádné dva prvky z  $A$  nejsou neporovnatelné, neboli pokud  $A$  lze uspořádat do jednoho řetězce.

Zkráceně se lineárnímu uspořádání říká jen *uspořádání*.

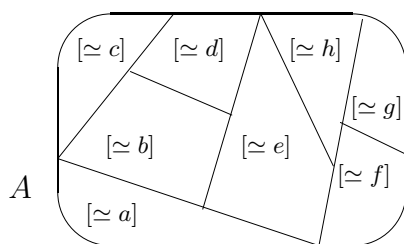
**Komentář:** Lineární uspořádání jsou například uspořádání celých či reálných čísel podle velikosti, nebo uspořádání všech slov slovníkovým způsobem. Důležité je, že pro každé dva různé prvky musíme umět rozhodnout, který z nich je větší.

## Ekvivalence

Příští definice je velice podobná Definici 4.4, jen vlastnost *antisymetrická* je nahrazena *symetrická*. Tato drobná úprava velice změní výsledné vyznění a místo porovnávání různých prvků získáme relaci, která “sdružuje” podobné prvky (Věta 4.6).

**Definice 4.5.** *Ekvivalence*  $\simeq$

je reflexivní, symetrická a tranzitivní binární relace na množině.



Obrázek 4.2: Schematické znázornění relace ekvivalence jako rozkladu množiny  $A$  na třídy ekvivalence.

**Definice:** Mějme relaci ekvivalence  $\simeq$  na množině  $A$ . *Třídou ekvivalence* prvku  $x$  rozumíme podmnožinu definovanou jako

$$[\simeq x] = \{z \in A : z \simeq x\}.$$

**Komentář:** *Příklady ekvivalencí:*

- Relace vyjadřující mezi studenty vztah “mají stejnou známku z DIM”.
- Relace vyjadřující vztah mezi léky “obsahují stejné účinné látky”.
- Relace “synonyma v mluveném jazyce” by také měla být ekvivalencí.

**Definice:** Říkáme, že podmnožiny  $X_1, X_2, \dots, X_m$  množiny  $Y$  tvoří *rozklad*  $Y$ , pokud jsou  $X_1, X_2, \dots, X_m$  po dvou disjunktní (tj. žádný prvek “se neopakuje”) a jejich sjednocení je úplné  $X_1 \cup X_2 \cup \dots \cup X_m = Y$ .

**Věta 4.6.** *Různé třídy ekvivalence  $\simeq$  na množině  $A$  tvoří rozklad  $A$ .*

**Důkaz:** Nejprve si uvědomme, že ve výčtu tříd ekvivalence všech prvků  $\{[\simeq x] : x \in A\}$  se stejné třídy mnohokrát opakují – přesněji řečeno, pro každou dvojici  $a \simeq b$  je  $[\simeq a] = [\simeq b]$  jako množiny. Pro důkaz faktu, že různé třídy (bez opakování) tvoří rozklad  $A$ , stačí ověřit toto

- $\bigcup_{x \in A} [\simeq x] = A$ ,
- a  $[\simeq x] \cap [\simeq y] = \emptyset$  pro každá  $x \neq y$ .

To první je zřejmé z definice tříd, neboť  $x \in [\simeq x]$  z reflexivity.

Druhou vlastnost dokážeme sporem – předpokládáme, že nějaké  $u \in [\simeq x] \cap [\simeq y]$ . To znamená, že  $u \simeq x$  a  $u \simeq y$ , což z tranzitivity a symetrie dává  $x \simeq y$  a tudíž  $[\simeq x] = [\simeq y]$ .  $\square$

## Úlohy k řešení

**(4.2.1)** Podívejme se na částečné uspořádání dělitelností čísel  $1, \dots, 12$  z Obrázku 4.1.

- a) Jaký je v něm nejdelší řetězec?
- b) Je v něm nejmenší či největší prvek?
- c) Kolik je tam maximálních prvků?

**(4.2.2)** Na systémy všech 5-prvkových podmnožin množiny  $\{1, 2, \dots, 12\}$  definujeme ekvivalenci následovně: Dvě množiny jsou ekvivalentní, pokud mají stejné nejmenší číslo. Kolik tříd ekvivalence získáme? Jak velká je největší třída?

## 4.3 Funkce, či zobrazení

Dalším význačným typem relace jsou zobrazení, která se většinou jako relace ani nevnímají, ale z formální podstaty relace stejně vycházejí. Přemýšlel čtenář někdy nad tím, jak správně popsat vztah mezi parametrem a výslednou hodnotou nějaké funkce? To je právě náplň tohoto oddílu. (Pro nás jsou v tomto textu slova “funkce” a “zobrazení” synonymy.)

**Definice 4.7.** *Zobrazení (funkce)* z množiny  $A$  do  $B$  je takovou podmnožinou kartézského součinu  $f \subseteq A \times B$ , ve které platí, že pro každý prvek  $x \in A$  existuje právě jediný prvek  $y \in B$  pro který  $(x, y) \in f$ .

Zobrazení zapisujeme obvykle symbolicky

$$f : A \rightarrow B$$

a druhý prvek dvojice jednoduše jako

$$y = f(x).$$

**Komentář:** Pro příklady zobrazení nemusíme chodit daleko:

- Klasické analytické funkce z  $\mathbb{R} \rightarrow \mathbb{R}$  jako  $y = x^2$ ,  $y = \sin(x)$ , ...  
Také funkce s více parametry třeba z  $\mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$  jako  $z = x + y$ ,  $z = 2x^{2y} \dots$
- Funkce ze všech studentů, kde každému je přiřazen jeho nejlepší kamarád.
- Funkce přiřazující jménu souboru v adresáři jeho umístění na disku (inode).

**Definice:** Funkce  $f : A \rightarrow B$  je

- *prostá (injektivní)* jestliže  $x \neq y \in A$  implikuje  $f(x) \neq f(y)$ ,
- *na (surjektivní)* jestliže pro každé  $z \in B$  existuje  $x \in A$ , že  $f(x) = z$ ,
- *vzájemně jednoznačná (bijektivní)* jestliže je prostá a na.

Bijektivní funkce budou pro nás mít zvláštní význam, jak si nyní přiblížíme.

**Poznámka** o isomorfismu:

Často se setkáváme s diskrétními strukturami, které jsou sice jinak prezentované či pojmenované, ale ve své podstatě jsou stejné – prvky jedné lze převést *bijekcí* na prvky druhé bez změny vlastností. To vyjadřujeme slovem *isomorfní*.

Například množina  $\{1, \dots, n\}$  má stejně mnoho podmnožin jako množina  $\{n + 1, \dots, 2n\}$ , protože mezi prvky existuje snadná bijekce předpisem  $b(i) = i + n$ . Co víc, i částečná uspořádání těchto systémů podmnožin inkluzí jsou si isomorfní přes rozšířenou bijekci  $b^*(X) = \{i + n : i \in X\}$ .

Jiným příkladem je třeba relace dělitelnosti na množině  $\{1, 2, 3, 4, 6, 9, 12, 18, 36\}$  — ta je isomorfní dělitelnosti na množině  $\{1, 2, 4, 5, 10, 20, 25, 50, 100\}$  v bijekci  $p$ , která v prvočíselném rozkladu nahradí prvočíslo 3 prvočíslem 5, třeba  $p(1) = 1$ ,  $p(3) = 5$ ,  $p(6) = 10$ ,  $p(9) = 25$ , ...

## 4.4 Skládání zobrazení; Permutace

Důležitým a užitečným atributem zobrazení je možnost jejich vzájemného skládání za sebou. To je definováno zcela přirozeně (jako už znáte u složených funkcí z analýzy).

**Definice:** Mějme zobrazení (funkce)  $f : A \rightarrow B$  a  $g : B \rightarrow C$ . Pak jejich *složením* rozumíme zobrazení  $(g \circ f) : A \rightarrow C$  definované

$$(g \circ f)(x) = g(f(x)).$$

**Poznámka:** Místo výše definovaného způsobu skládání zobrazení se také často píše složení jako  $(fg)$  a skládá se “naopak”  $(fg)(x) = g(f(x))$ . Proto je třeba vždy ujasnit, který způsob skládání používáme!

Pro zbytek tohoto oddílu se zaměříme na permutace coby zobrazení.

**Fakt:** Nechť *permutace*  $\pi$  na množině  $[1, n]$  je určena seřazením  $(p_1, \dots, p_n)$ . Pak  $\pi$  je také *bijektivním zobrazením*  $[1, n] \rightarrow [1, n]$  definovaným předpisem  $\pi(i) = p_i$ . Navíc lze permutace skládat jako zobrazení podle výše uvedené definice.

**Fakt:** Všechny permutace množiny  $[1, n]$  spolu s operací skládání tvoří grupu, zvanou *symetrická grupa*  $S_n$ .

**Poznámka:** Permutační grupy (podgrupy grupy všech permutací) jsou velice důležité v algebře, neboť každá grupa je isomorfní některé permutační grupě.

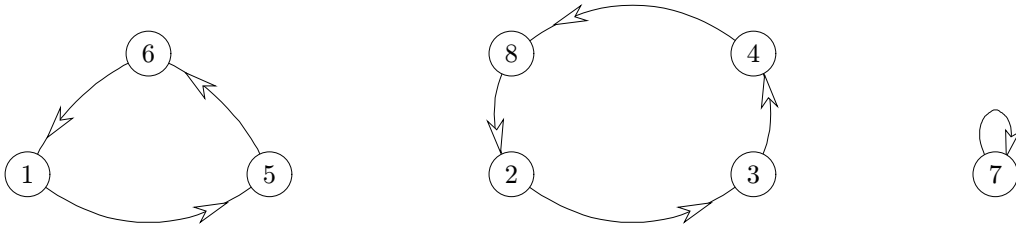
V kontextu pohledu na skládání zobrazení si zavedeme jiný, vhodnější, způsob zápisu permutací – pomocí jejich cyklů.

**Definice:** Necht  $\pi$  je permutace na množině  $A$ . *Cyklem* v  $\pi$  rozumíme posloupnost  $(a_1, a_2, \dots, a_k)$  různých prvků  $A$  takovou, že  $\pi(a_i) = a_{i+1}$  pro  $i = 1, 2, \dots, k-1$  a  $\pi(a_k) = a_1$ .

Přitom, jak název napovídá, v zápise cyklu  $(a_1, a_2, \dots, a_k)$  není důležité, kterým prvkem začneme, ale jen dodržení cyklického pořadí. Cyklus v permutaci může mít i jen jeden prvek (zobrazený na sebe).

**Komentář:** Nakreslete si (vámi zvolenou) permutaci  $\pi$  obrázkem, ve kterém vedete šipku vždy od prvku  $i$  k prvku  $\pi(i)$ . Pak uvidíte, že cykly dle naší definice jsou právě cykly tvořené šipkami ve vašem obrázku. S tímto grafickým zobrazením pro vás nebude problém pochopit následující tvrzení.

Například permutaci  $5, 3, 4, 8, 6, 1, 7, 2$  si lze obrázkem nakreslit takto:



**Lema 4.8.** Každou permutaci na konečné množině  $A$  lze zapsat jako složení cyklů na disjunktních podmnožinách  $A$ .

**Důkaz:** Vezmeme libovolný prvek  $a_1 \in A$  a iterujeme zobrazení  $a_2 = \pi(a_1)$ , atd., až se dostaneme zpět k  $a_1$ . (To se musí stát, jelikož je  $A$  konečná.) Tak získáme první cyklus  $(a_1, \dots, a_k)$ . Stejně pokračujeme s hledáním dalších cyklů ve zbylé množině  $A \setminus \{a_1, \dots, a_k\}$ .  $\square$

**Značení permutací cykly:** Necht se permutace  $\pi$  podle Lematu 4.8 skládá z cyklů  $(a_1, \dots, a_k)$ ,  $(b_1, \dots, b_l)$  a třeba  $(c_1, \dots, c_m)$  (nebo i libovolný jiný počet cyklů). Pak zapíšeme

$$\pi = (a_1, \dots, a_k)(b_1, \dots, b_l)(c_1, \dots, c_m).$$

**Komentář:** Pro ilustraci. Vezmeme 7-prvkovou permutaci  $5, 3, 4, 2, 6, 1, 7$ . Ta se rozkládá na tři cykly  $(1, 5, 6)$ ,  $(2, 3, 4)$  a  $(7)$ . Jiná permutace  $3, 4, 5, 6, 7, 1, 2$  má jediný cyklus  $(1, 3, 5, 7, 2, 4, 6)$ .

Nyní určíme složení těchto dvou permutací (zápisem cykly):

$$(1, 5, 6)(2, 3, 4)(7) \circ (1, 3, 5, 7, 2, 4, 6) = (1, 4)(2)(3, 6, 5, 7)$$

(Nezapomínejme, že první se ve složení aplikuje pravá permutace! Postup skládání jsme použili následovně: 1 se zobrazí v permutaci vpravo na 3 a pak vlevo na 4. Dále 4 se zobrazí na 6 a pak na 1. Tím jsme uzavřeli první cyklus  $(1, 4)$ . Dále se 2 zobrazí na 4 a pak hned zpět na 2, tj. má samostatný cyklus. Zbýlý cyklus určíme analogicky.)

## Úlohy k řešení

**(4.4.1)** V zápise permutací cykly nalezněte složení těchto dvou permutací:

$$(1, 2, 5, 6)(3, 4) \circ (1, 3, 4, 5, 6)(2).$$

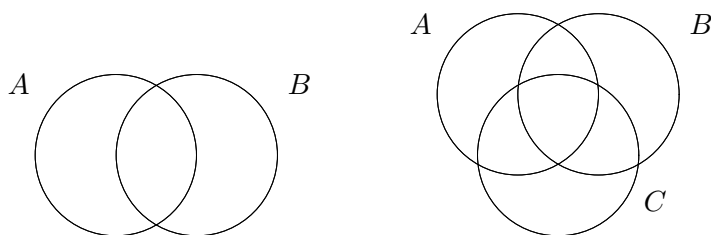
(4.4.2) Kolikrát musíme permutaci  $(1, 2, 3)(4, 5)$  složit samu se sebou, abychom dostali identickou permutaci? (Která zobrazí každé číslo na sebe sama.)

\*(4.4.3) Kolik je permutací  $n$ -prvkové množiny s jediným cyklem?

## 4.5 Princip inkluze a exkluze

(Někdy také nazýván “princip zapojení a vypojení”...)

Následující důležitý množinový princip často využíváme v jednoduchých aplikacích, aniž si to uvědomujeme.



Obrázek 4.3: Průnik dvou a tří množin – pro ilustraci inkluze a exkluze.

**Věta 4.9.** Počet prvků ve sjednocení dvou či tří množin spočítáme:

$$|A \cup B| = |A| + |B| - |A \cap B|$$

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$$

**Příklad 4.10.** Podívejme se na typické použití principu inkluze a exkluze.

Mezi studenty má 15 na svém počítači M\$ Windows a 17 Linux. Přitom 6 z nich má oba systémy zároveň. Kolik studentů má aspoň jeden z těchto systémů?

$$|W \cup L| = |W| + |L| - |W \cap L| = 15 + 17 - 6 = 26$$

Nechť ještě navíc 5 studentů má na svém počítači FreeBSD. Přitom jeden student má všechny tři systémy dohromady a další 2 mají Linux a FreeBSD najednou. Kolik studentů má aspoň jeden z těchto tří systémů?

$$\begin{aligned} |W \cup L \cup B| &= |W| + |L| + |B| - |W \cap L| - |W \cap B| - |B \cap L| + |W \cap L \cap B| = \\ &= 15 + 17 + 5 - 6 - 1 - 3 + 1 = 28 \end{aligned}$$

□

**Komentář:** Zájemci o doplňující studium si mohou promyslet obecný princip inkluze a exkluze pro více množin, jak je symbolicky zapsaný zde:

$$\left| \bigcup_{j=1}^n A_j \right| = \sum_{\emptyset \neq I \subseteq \{1, \dots, n\}} (-1)^{|I|-1} \cdot \left| \bigcap_{i \in I} A_i \right|$$

(Jeho znalost nebude v předmětu vyžadována.)

### Úlohy k řešení

(4.5.1) Mezi 30 žáky ve třídě jich 10 hraje fotbal a 20 dělá jiný sport. Z toho ale 5 žáků hraje současně fotbal a ještě dělá jiný sport. Kolik žáků celkem sportuje?

(4.5.2) Ze 1000 televizí jich při první kontrole na výrobní lince má 5 vadnou obrazovku, 10 je poškrábaných a 12 má jinou vadu. Přitom 3 televize mají současně všechny tři vady a 4 jiné jsou poškrábané a mají jinou vadu. Kolik televizí je celkem vadných?

### Rozšiřující studium

Problematika relací, funkcí a uspořádání je obsáhle popsána ve většině učebnic algebry. Pro naše účely stačí zjednodušený pohled, který je prezentován třeba v [5, Oddíly 1.4,5,6,7]. Princip inkluze a exkluze (který je v kombinatorice velmi důležitý) je obsáhle popsán v [5, Oddíl 2.6].

## 4.6 Cvičení: Příklady na relace a jiné

**Příklad 4.11.** Kutil Tom má v dílně strašný nepořádek. V jedné bedýnce má pomíchané šroubky a matky všech možných průměrů. Definujme si nyní relaci  $P$ , ve které je každý šroubek a každá matka v relaci sám se sebou (reflexivní). Navíc jsou v relaci  $P$  symetricky všechny dvojice šroubek–matka, které do sebe přesně pasují. Zdůvodněte, proč relace  $P$  nemusí být v obecnosti relací ekvivalence. Za jakých dodatečných podmínek je  $P$  relací ekvivalence?

Definovaná relace je reflexivní a symetrická. Problémy jsou však s tranzitivitou. Dva různé šroubky mohou pasovat do stejné matky, takže by podle tranzitivity měly také být v relaci, ale tak tomu není. Stejný problém nastává se dvěma matkami pasujícíma na stejný šroubek.

Pro druhou část se podívejme, jak výše uvedené problémy můžeme vyloučit. Stačí požadovat, aby k jednomu šroubku pasovala nejvýše jedna matka a naopak. Pak definovaná relace bude i tranzitivní a bude to ekvivalence.  $\square$

**Příklad 4.12.** Na množině všech přirozených čísel od 0 do 1000 definujeme binární relaci  $R$  následovně:  $(a, b) \in R$  právě když platí

a)  $-1 \leq a^2 - b^2 \leq 4$ ,

b)  $0 \leq a - b \leq 1$  nebo číslo  $a$  je trojnásobkem čísla  $b$ ,

c)  $0 \leq a - b \leq 1$  nebo číslo  $a$  je polovinou čísla  $b$ ,

Odpovězte, jaké vlastnosti relace  $R$  má: je reflexivní, symetrická, antisymetrická, tranzitivní? Pokud některou z těchto vlastností relace  $R$  nemá, napište konkrétními čísly protipříklad.

a) Není těžké zjistit, že v relaci jsou všechny dvojice  $(x, x)$ , tedy je to reflexivní relace. Dále zjistíme, že mimo reflexivních dvojic jsou v relaci  $R$  už jen dvojice  $(0, 1)$ ,  $(1, 0)$ ,  $(2, 0)$ ,  $(2, 1)$ . Hrubou silou (probráním těchto pár dvojic) pak ověříme, že relace je i tranzitivní. Pro protipříklad, že  $R$  není symetrická, máme  $(2, 0) \in R$ , ale  $(0, 2) \notin R$ . Jelikož dále  $(0, 1)$ ,  $(1, 0) \in R$ , ale  $0 \neq 1$ , není  $R$  ani antisymetrická.

b) Tato relace je opět reflexivní. Jinak  $(9, 3)$ ,  $(3, 1) \in R$ , ale  $(9, 1) \notin R$  ani  $(1, 3) \notin R$ , takže  $R$  není tranzitivní ani není symetrická. Po chvilce zkoumání také přijdeme na to, že z  $(a, b) \in R$  vyplývá  $a \geq b$ , takže  $R$  je antisymetrická.

c) Tato relace je opět reflexivní. Žádnou z dalších jmenovaných vlastností nemá, jak zjistíme z toho, že  $(1, 2)$ ,  $(2, 1) \in R$ , ale  $1 \neq 2$ , dále  $(3, 2) \in R$ , ale  $(2, 3)$ ,  $(3, 1) \notin R$ .  $\square$

**Příklad 4.13.** 8 děvčat a 5 chlapců sedí okolo kulatého stolu rozesazeni tak, že čtyři děvčata jsou vedle sebe a jinak se pak střídají chlapec–děvče. Na povel se vymění podle následujícího schématu: Každý chlapec se přesune na místo nejbližšího chlapce po jeho pravici, každé děvče se přesune na místo nejbližšího děvčete po jeho levici. Po kolika opakováních této výměny se vrátí rozesazení u stolu přesně do výchozího stavu? Zdůvodněte.

Ze zadání se jedná o permutaci přisedících u stolu, která má právě dva cykly – jeden o 5 chlapcích a druhý o 8 děvčatech. Chlapci se dostanou na svá místa vždy po násobku

pěti přesunech, děvčata po počtu přesunů, který je násobkem osmi. Jelikož 5 a 8 jsou nesoudělná, nejmenší společný násobek je  $5 \cdot 8 = 40$  přesunů. (Při řešení vůbec nezáleží, jak vlastně u stolu seděli, důležitá je jen existence dvou cyklů v permutaci.)  $\square$

**Příklad 4.14.** *Kolika nulami končí desítkový zápis  $33!$  ?*

Tento příklad nezapadá do žádného zatím probraného schématu, ale má své kouzlo a vztah k diskrétní matematice. Nepokoušejte se číslo  $33!$  spočítat na vaší kalkulačce, její rozsah na to nestačí. Místo toho se zamysleme, jak v součinu  $1 \cdot 2 \cdot \dots \cdot 32 \cdot 33$  vznikají nuly na konci. Jistě víte, že číslo 10 má prvočíselný rozklad  $2 \cdot 5$ , takže jakmile se v našem součinu objeví v rozkladu prvočísla 2 a 5, na konci za ně přibude jedna nula.

Úkol tedy vlastně zní, spočítat, kolikrát se v prvočíselném rozkladu součinu  $1 \cdot 2 \cdot \dots \cdot 32 \cdot 33$  vyskytují prvočísla 2 a 5. Zřejmě se prvočíslu 2 vyskytuje mnohem častěji, takže budeme počítat pětky. Ty jsou po jedné v činitelích 5, 10, 15, 20, dvě v 25 a opět jedna v 30. To je vše. Takže číslo  $33!$  končí  $4 + 2 + 1 = 7$  nulami.  $\square$

**Úlohy k řešení**

**(4.6.1)** *Na množině všech slov-řetězců nad běžnou abecedou definujeme relaci následovně: Slovo  $x$  je v relaci se slovem  $y$  pokud je  $y$  obsaženo jako podřetězec v  $x$ . O jaký známý typ relace se jedná? Zdůvodněte.*

**(4.6.2)** *Na množině všech slov-řetězců nad běžnou abecedou definujeme relaci následovně: Slovo  $x$  je v relaci se slovem  $y$  pokud lze  $x$  získat z  $y$  jen přeházením písmen. O jaký známý typ relace se jedná? Zdůvodněte.*

**(4.6.3)** *Mezi všemi studenty VŠB (obojího pohlaví) definujeme následující relaci: Dva studenti jsou v relaci, pokud mají*

- a) stejný rok i stejný měsíc narození,
- b) stejný rok narození nebo různé pohlaví,
- c) stejný měsíc narození nebo stejné pohlaví,
- d) stejný rok narození a stejné pohlaví.

*Jaké vlastnosti má tato relace (jako reflexivní, symetrická, antisymetrická, tranzitivní)? Jedná se třeba o částečné uspořádání nebo o ekvivalenci?*

**\*(4.6.4)** *Mezi všemi studenty na přednášce DIM definujeme následující binární relaci: Každý student je v relaci sám se sebou, tj. je to reflexivní relace. Každý student  $A$  je v relaci s jiným studentem  $B$ , právě když  $B$  sedí*

- a) ve stejné řadě nalevo od  $A$ ,
- b) ve stejné řadě jako  $A$  nebo v řadě hned za  $A$ ,
- c) v poslední řadě (nezáleží, kde sedí  $A$ ),
- d) v některé řadě před  $A$ .

*Jaké vlastnosti má tato relace (jako symetrická, antisymetrická, tranzitivní)? Jedná se třeba o částečné uspořádání nebo o ekvivalenci? Vaši odpověď dobře zdůvodněte.*

**\*(4.6.5)** *Na množině všech přirozených čísel od 0 do 1000 definujeme binární relaci  $R$  následovně:  $(a, b) \in R$  právě když číslo  $(a + 2b)$  je dělitelné třemi. Zjistěte, jaké vlastnosti relace  $R$  má. Jedná se třeba o ekvivalenci nebo uspořádání?*

**(4.6.6)** *Z kolika cyklů se skládá tato permutace?  $7, 2, 3, 6, 4, 5, 1$*

**(4.6.7)** *Složte dvě permutace  $(1, 4, 6, 5, 7)(2)(3) \circ (1, 7)(2)(3)(4, 6, 5)$  v zápise cykly.*

**(4.6.8)** *Složte dvě permutace  $(1, 4, 5, 6, 7)(2)(3) \circ (1, 7)(2,3)(4, 6, 5)$  v zápise cykly.*

**(4.6.9)** *Složte dvě permutace  $(1, 2, 3, 4, 5)(6, 7, 8) \circ (2, 3, 4, 5, 6, 7)(1, 8)$  v zápise cykly.*

**(4.6.10)** *Mějme nějaké dvě permutace množiny  $[1, 9]$ , každou složenou z jediného cyklu. Kolik nejvíce cyklů může obsahovat permutace vzniklá jejich složením?*

**(4.6.11)** *Kolika nulami končí desítkový zápis  $66!$  ?*

**(4.6.12)** *Jaká je poslední nenulová číslice  $33!$  ?*

Návod: Obdobně Příkladu 4.14 vytkněte z rozkladu  $33!$  prvočinitele  $2 \cdot 5$  (jen v páru!) a ze zbylých činitelů stačí vynásobit poslední číslice.



\*(4.6.13) Jaká je poslední nenulová číslice ve “složeném” kombinačním čísle

$$\binom{\binom{10}{5}}{\binom{5}{2}}?$$

## 5 Algoritmizace diskrétních struktur

### Úvod

Na závěr první části našeho učebního textu zařazujeme doplňkovou lekci, která neformálně popisuje programátorskou implementaci některých dříve prezentovaných diskrétních struktur a konceptů. Ukážeme, jak lze implementovat posloupnosti, množiny, relace a zobrazení, jak je možno počítačově generovat různé typy výběrů, atd. Také se zmíníme o generátorech náhodných čísel.

Ukázky programových kódů zde nejsou vyčerpávající ani jediné možné, jsou to jen jednoduché návody, jak by se s předvedenými strukturami mohlo v počítačových programech pracovat. (Samozřejmě mohou existovat jiné a třeba i lepší implementace, než jsou popsány zde.)

### Cíle

Cílem této lekce je ukázat čtenáři praktické programové implementace některých diskrétních struktur a běžných základních algoritmů nad nimi. Jedná se skutečně jen o základní příklady, nikoliv o vyčerpávající přehled.

### 5.1 Programová implementace struktur

Nejprve se podíváme na implementace základních struktur posloupnosti, zobrazení a obecné binární relace. (Implementaci množin pro jejich širší záběr ponecháme do samostatného oddílu.)

**Posloupnost**  $(a_0, a_1, \dots, a_n)$  implementujeme obvykle jako jednorozměrné pole  $a[\ ]$ , kde  $a[i] = a_i$ .

**Zobrazení**  $f : A \rightarrow B$  implementujeme třeba obdobně jako posloupnost – pokud  $A = \{a_1, \dots, a_n\}$  a  $B = \{b_1, \dots, b_m\}$ , použijeme pole  $f[\ ]$ , ve kterém  $f[i]=j$  vyjadřuje  $f(a_i) = b_j$ .

**Poznámka:** Taková implementace se velmi hodí, pokud přímo zobrazujeme čísla, pro jiné objekty však ještě musíme “překládat” prvky z  $A$  či  $B$  na jejich indexy, což může být obtížné. Pak se lépe hodí různé pokročilé datové typy jako hašovací tabulky, atd... (Viz také implementace množin.)

**Binární relace**  $R$  na množině  $A$  se nejpřirozeněji implementuje pro  $A = \{a_1, \dots, a_n\}$  dvourozměrným polem (maticí)  $r[\ ][\ ]$ , ve kterém je  $r[i][j]=0$  pokud  $(a_i, a_j) \notin R$  a  $r[i][j]=1$  pokud  $(a_i, a_j) \in R$ .

**Příklad 5.1. Permutace:** Pro ukázky operací se zobrazeními se soustředíme v příkladech na permutace, které lze vidět jako bijektivní zobrazení  $p[\ ]$  množiny  $\{0, 1, \dots, n-1\}$  na sebe.

a) Otestovat, zda zobrazení  $p[\ ]$  je skutečně permutace, můžeme rychle takto (vlastně jen zjišťujeme, zda zobrazení je na):

```

for (i=0; i<n; i++) u[i] = 0;
for (i=0; i<n; i++) u[p[i]] = 1;
for (i=0; i<n; i++) if (u[i]!=1)
    printf("Neni permutace!");

```

(Pole `u[]` značí, na který prvek se už něco zobrazilo.)

- b) Skládání dvou permutací `p[]`, `q[]` jako zobrazení se realizuje:

```

for (i=0; i<n; i++)
    r[i] = q[p[i]];

```

- c) Výpis všech cyklů  $n$ -prvkové permutace `p[]` získáme třeba kódem:

```

for (i=0; i<n; i++) u[i] = 0;
for (i=0; i<n; i++) if (u[i]==0) {
    printf("\ncyklus: %d",i); u[i] = 1;
    for (j=p[i]; j!=i; j=p[j])
        { printf(",%d",j); u[j] = 1; }
}

```

(Pro vysvětlení algoritmu, pole `u[]` zde značí, který prvek jsme již v některém cyklu vypisali. Druhá smyčka vždy hledá první nevypsany prvek  $x$  a pak začíná vypisovat iterativně nový cyklus obsahující  $x$ , dokud se iterací zobrazení `p[]` nedostane zpět do  $x$ .)  $\square$

**Příklad 5.2. Relace:** Mějme pro tento příklad binární relaci  $R$  na množině  $\{0, 1, \dots, n-1\}$ , reprezentovanou dvourozměrným polem `r[][]` jako výše.

- a) Otestovat, zda relace  $r$  je symetrická, je velmi jednoduché:

```

for (i=0; i<n; i++) for (j=i+1; j<n; j++)
    if (r[i][j]!=r[j][i]) {
        printf("Neni symetricka!");
        return -1;
    }

```

- b) Test tranzitivity dané relace lze provést třeba takto:

$$\forall i, j, k : r[i][j] \wedge r[j][k] \Rightarrow r[i][k]$$

```

for (i=0; i<n; i++) for (j=0; j<n; j++) {
    if (!r[i][j]) continue;
    for (k=0; k<n; k++) {
        if (!r[j][k]) continue;
        if (r[i][k]) continue;
        printf("Neni tranzitivni!");
        return -1;
    }
}

```

- Další základní operace se zobrazeními nebo relacemi by se implementovaly analogicky, nemělo by to, s použitím zdejších příkladů, činit velké potíže.  $\square$

## 5.2 Implementace množin

Množiny se ze všech základních struktur implementují (kupodivu) nejobtížněji. To plyne z jejich principiální neuspořádanosti – nevíme totiž, kde “který prvek uložit” a kde jej pak zase jako prvek množiny najít. Proto se používají různé postupy jak s pevným umístěním prvků (kdy se však velmi plýtvá pamětí pro potenciální prvky, které zatím prvky množiny nejsou), tak i s proměnným umístěním prvků (kdy je zase obtížné prvek v množině vyhledat).

### Charakteristická funkce podmnožiny:

Předpokladem je, že předem známe celé *univerzum*  $U$ , ze kterého jsou naše prvky vybírány. Nechť  $U = \{u_1, \dots, u_n\}$ , pak podmnožinu  $X \subseteq U$  implementujeme jako

pole  $x[ ]$ , kde  $x[i]=1$  pro  $u_i \in X$  a  $x[i]=0$  jinak.

Zde se velmi snadno hledají prvky množiny, sjednocení je funkcí *OR*, průnik funkcí *AND*. . . Velkou nevýhodou je však použitelnost jen **pro malá univerza**  $U$ !

### Seznam prvků množiny:

Množinu  $X$  implementujeme jednoduchým seznamem všech jejích prvků. Pokud je seznam prvků  $X$  uložen v poli  $x[ ]$ , pak můžeme psát

$X = \{x[1], x[2], \dots, x[k]\}$  pro pole  $x[ ]$  délky  $k$ .

Implementace je vhodná i pro velmi velká a předem neurčená univerza. Často je lepší místo pole použít *dynamický spojový seznam prvků*, pak lze snadno přidávat a vypouštět prvky kdekoliv v seznamu. Nevýhodou je, že vyhledání prvku množiny (obvykle nejčastější požadovaná operace) je zdlouhavé – musí se projít celý seznam.

### Uspořádaný seznam prvků množiny

Jedná se o variantu předchozí implementace, kdy prvky v seznamu jsou navíc lineárně uspořádány podle předem daného klíče, třeba dle velikosti v případě čísel, nebo dle abecedy u jmen, atd. Hlavním přínosem je možnost *binárního vyhledávání* prvků v množině metodou půlení intervalu v seznamu (viz Příklad 5.4).

Pokročilé datové struktury jako stromy, haldy, hashovací tabulky a jiné už nepatří do této lekce. . .

**Příklad 5.3.** Sjednocení dvou množin daných poli prvků  $a[ ]$ ,  $b[ ]$  do pole  $c[ ]$ :

```
for (i=0; i<n; i++)
    c[i] = a[i];
for (i=0, k=n; i<m; i++) {
    for (j=0; j<n; j++)
        if (b[i]==a[j]) break;
    if (j<n) continue;
    c[k++] = b[i];
}
```

□

**Příklad 5.4.** Binární vyhledávání čísla  $k$  v uspořádaném poli  $p[ ]$  délky  $n$ :

```
a = 0; b = n-1;
while (a<b && p[a]!=k) {
    c = (a+b)/2;
    if (p[c]<k) a = c+1;
    else b = c;
}
if (p[a]!=k) printf("Cislo k neni v seznamu.");
```

Všimněme si, že oproti běžnému vyhledávání projitím všech prvků zde uděláme pouze  $\lceil \log_2 n \rceil$  vyhledávacích kroků. □

### 5.3 Generování výběrů

Častým programátorským úkolem je procházet všechny možné výběry určitého typu. My si zde ukážeme netriviální postupy, jak procházet zobrazeními, variacemi a kombinacemi z předem daných prvků.

**Jednoduché procházení** dvojic (trojic, atd...):

Všechny uspořádané dvojice indexů  $i, j$  projdeme jednoduše dvojí smyčkou

```
for (i=0; i<N; i++) for (j=0; j<N; j++) ...
```

(kde někdy musíme ošetřit zvlášť případ  $i=j$ ).

Všechny neuspořádané dvojice projdeme podobnou smyčkou

```
for (i=0; i<N; i++) for (j=i+1; j<N; j++) ...
```

**Procházení všech zobrazení**

**Příklad 5.5.** Všechna  $n^k$  zobrazení  $k$ -prvkové do  $n$ -prvkové množiny

$$\text{map} : \{0, 1, \dots, K - 1\} \rightarrow \{0, 1, \dots, N - 1\}$$

projdeme jednoduše kódem:

```
int    i, j, map[K];
map[i = 0] = -1;
while (i >= 0) {
    if (++map[i] >= N)
        { i--; continue; }
    if (++i < K)
        { map[i] = -1; continue; }
    // zpracujeme zobrazení i -> map[i]
    i--;
}
```

Podívejme se blíže na tento kód, který vlastně emuluje rekurzivní volání výběrů jednotlivých prvků (je to tak výrazně efektivnější). Jinými slovy, místo abychom volili smyčkou obraz prvního prvku  $\text{map}[0]$ , pak se zanořili do rekurze a v ní volili obraz druhého prvku  $\text{map}[1]$ , atd., tak se pohybujeme v jednom cyklu v proměnné “úrovni vnoření”  $i$  (od 0 do  $k - 1$ ) a vybíráme vždy obraz  $\text{map}[i]$ .

Pro každou volbu  $\text{map}[i]$  testujeme:

- Zda už překročila rozsah  $N$ , pak se vracíme na předchozí úroveň.
- Zda už byla poslední volba  $K$ -tého obrazu, jinak na další úroveň.

□

**Procházení všech variací**

**Příklad 5.6.** Všechny  $k$ -prvkové variace (bez opakování) z  $n$  prvků projdeme následovně:

```
int    i, j, select[K];
select[i = 0] = -1;
while (i >= 0) {
    if (++select[i] >= N)
        { i--; continue; }
    for (j=0; j<i; j++)
        if (select[i]==select[j]) break;
    if (j<i) continue;
    if (++i < K)
        { select[i] = -1; continue; }
```

```

        // zpracujeme variaci (select[0],...,select[K-1])
        i--;
    }

```

Obdobně Příkladu 5.5; pro každou volbu `map[i]` zde testujeme:

- Zda už překročila rozsah  $N$ , pak se vracíme na předchozí úroveň.
- Zda se neopakuje předchozí prvek volby, jinak výběr přeskočíme.
- Zda už byla poslední volba  $K$ -tého obrazu, jinak na další úroveň.

□

## Procházení všech kombinací

**Příklad 5.7.** Projdeme všechny  $k$ -prvkové kombinace (bez opakování) z  $n$  prvků.

Tento algoritmus je velice podobný předchozímu Příkladu 5.6, jenomže nyní všechny výběry už generujeme jednoznačně seřazené podle velikosti, a proto každá kombinace vyjde jen jednou (a ne v různých pořadích jako výše u variací).

```

int    i,j, select[K];
select[i = 0] = -1;
while (i>=0) {
    if (++select[i]>=N)
        { i--; continue; }
    if (++i<K) {
        select[i] = select[i-1]; /* zde seřazene! */
        continue;
    }

    // zpracujeme kombinaci (select[0],...,select[K-1])
    i--;
}

```

(Všimněme si, že už není třeba testovat opakované výběry, protože prvky generované kombinace jsou ostře seřazené.) □

## 5.4 Generátory náhodných čísel

Neboli skutečné generování náhodné posloupnosti bitů v počítači.

**Komentář:** Kde všude potřebujeme používat náhodná čísla/bitů?

- Vytváření náhodných (velkých) privátních klíčů, třeba v SSL certifikátech.
- Použití náhodného hesla při šifrování SSL přenosu (zde musí být heslo skutečně náhodné, jinak by jej mohl nepřítel uhodnout).
- Řešení kolizí paketů na ethernetu náhodně dlouhým čekáním s příštím vysláním.
- Využití při *pravděpodobnostních algoritmech*, které dokážou mnohdy využít náhodné bity pro výrazné statistické urychlení běhu výpočtu.
- Při statistické analýza algoritmů a reálných procesů, různé modelování chaotických fyzikálních dějů, atd.

## Typy náhodných generátorů

### *Primitivní pseudonáhodné* generátory

využívají aritmetické vzorce typu

$$x := (A \cdot x + B) \bmod C,$$

které se pořád dokola iterují a vybrané bity  $x$  vytvářejí posloupnost.

Nevýhody – *velká závislost* bitů na sobě a *snadná předvídatelnost*.

### *Pseudonáhodné* generátory s vnějším vstupem

využívají podobné vzorce jako předchozí typ, ale navíc přidávají vstupy z *okolních fyzických procesů* (jako prodlevy stisků kláves, zpoždění operací disku, statistika síťových paketů, atd.).

Problémy – závislost na okolí a *ovlivnitelnost* vnějšími podmínkami, také velká “cena” každého bitu.

### *Hardwarové náhodné* generátory

jsou založeny na různých *kvantových šumech* (třeba přechodové stavy polovodičů), které se převádějí na posloupnost bitů.

Problémy – jak převést šum na posloupnost uniformních bitů, také jaká je naše důvěra v kvantovou mechaniku.

**Komentář:** Zkuste si v Linuxu spustit `'cat /dev/random'` a sledovat, co se stává po stiscích kláves či hýbání myši...

O jaký typ generátoru se asi jedná?

## 5.5 Kombinatorická “exploze”

Při programátorském řešení problémů spjatých s diskretní matematikou často používáme algoritmy ve stylu *projdí všechny možnosti a podle nich rozhodni*. V takových případech však často narážíme (a hodně tvrdě!) na fenomén zvaný *exponenciální kombinatorická exploze* prohledávaných možností. Jen se podívejte, jak rychle roste funkce faktoriál. (Mimochodem, tento úkaz je už známý z dávné historie – vzpomeňme si na pohádkový příběh se zrnky obilí na polích šachovnice...)

Matematicky popsáno, počet prohledávaných možností obvykle roste nějakou “divokou” exponenciální funkcí, a tudíž při zvětšení vstupu o 1 se potřebný čas výpočtu mnohonásobí. Pak se například stává, že výpočet pro vstup o velikosti 10 zvládne každý starý šrot, ale vstup o velikosti 15 **nelze spočítat** ani na nejvýkonnějších počítačích na světě.

*Takže pozor, mějte tento fenomén na paměti, až budete programovat procházení možností “hrubou silou”!*

### **Rozšiřující studium**

Pokud má čtenář zájem o širší popis algoritmické implementace diskretních struktur, doporučujeme knihu L. Kučery [3] nebo nový interaktivní projekt [4]. Asi nejrozsáhlejší učebnicí věnující se implementaci struktur a algoritmů je Knuthova [2] v angličtině.

# Část II

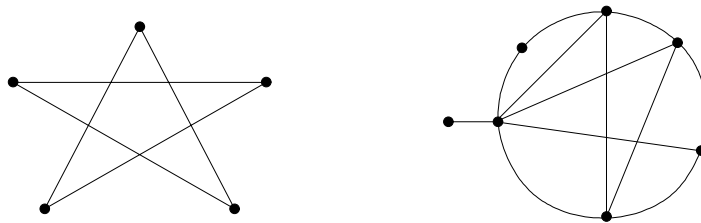
## Úvod do Teorie Grafů

### 6 Pojem grafu

#### Úvod

Třebaže grafy jsou jen jednou z mnoha struktur v diskrétní matematice, vydobily si svou užitečností a názorností tak důležité místo na slunci, až se dá bez nadsázky říci, že teorie grafů je asi nejdůležitější součástí soudobé diskrétní matematiky. Proto se i větší část našeho učebního textu věnuje grafům a jejich praktickému použití.

Neformálně řečeno, graf se skládá z vrcholů (představme si je jako nakreslené puntíky) a z hran, které spojují dvojice vrcholů mezi sebou. (Pozor, **nepleťme si graf s grafem funkce!**)



Obrázek 6.1: Úvodní ukázky grafů.

Takový graf může vyjadřovat souvislosti mezi objekty, návaznosti, spojení nebo toky, atd. Svě důležité místo v informatice si grafy získaly dobře vyváženou kombinací svých vlastností – snadným názorným nakreslením a zároveň jednoduchou zpracovatelností na počítačích. Díky těmto vlastnostem se grafy prosadily jako vhodný matematický model pro popis různých, i komplikovaných, vztahů mezi objekty.

#### Cíle

Prvním cílem této lekce je formálně definovat, co je to graf a jaké jsou nejzákladnější grafové pojmy. Také jsou zde uvedeny některé obvyklé typy grafů, jako cesty, kružnice, či úplné grafy. Především je však důležité, aby čtenář pochopil pojem isomorfismu grafů a dobře si jej procvičil v následujícím cvičení.

#### 6.1 Definice grafu

Hned na úvod přistoupíme k formální definici grafu. Bude se jednat o základní definici tzv. obyčejného grafu; přičemž možné rozšiřující definice zmíníme krátce v závěru, ale stejně se budeme převážně zabývat obyčejnými grafy. V základní definici popisujeme hrany mezi dvojicemi vrcholů pomocí dvouprvkových podmnožin těchto vrcholů.

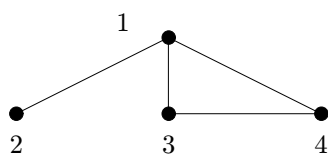
**Definice 6.1.** **Graf** (rozšířeně *obyčejný* či *jednoduchý neorientovaný* graf) je uspořádaná dvojice  $G = (V, E)$ , kde  $V$  je množina *vrcholů* a  $E$  je množina *hran* – množina vybraných dvouprvkových podmnožin množiny vrcholů.

**Značení:** Hranu mezi vrcholy  $u$  a  $v$  píšeme jako  $\{u, v\}$ , nebo zkráceně  $uv$ . Vrcholy spojené hranou jsou *sousední*. Na množinu vrcholů známého grafu  $G$  odkazujeme jako na  $V(G)$ , na množinu hran  $E(G)$ .

**Fakt:** Na graf se lze dívat také jako na symetrickou ireflexivní relaci, kde hrany tvoří právě dvojice prvků z této relace.

Poznámka: Grafy se často zadávají přímo názorným obrázkem, jinak je lze také zadat výčtem vrcholů a výčtem hran. Například:

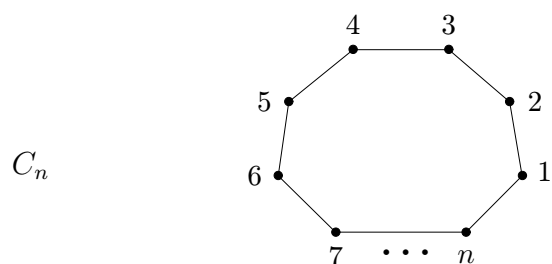
$$V = \{1, 2, 3, 4\}, \quad E = \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{3, 4\}\}$$



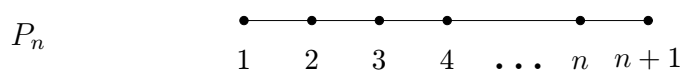
### Běžné typy grafů

Pro snadnější vyjadřování je zvykem některé běžné typy grafů nazývat popisnými jmény.

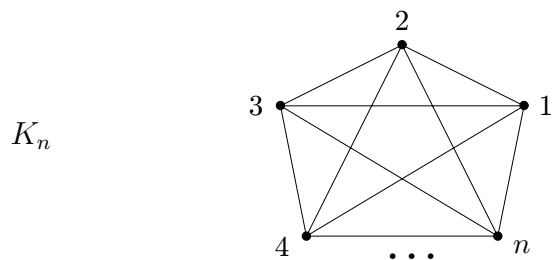
**Kružnice délky  $n$**  má  $n \geq 3$  vrcholů spojených do jednoho cyklu  $n$  hranami:



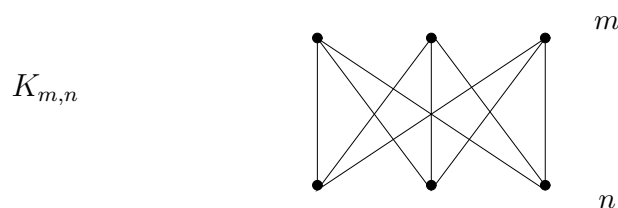
**Cesta délky  $n$**  má  $n + 1$  vrcholů spojených za sebou  $n$  hranami:



**Úplný graf** na  $n \geq 1$  vrcholech má  $n$  vrcholů, všechny navzájem pospojované (tj. celkem  $\binom{n}{2}$  hran):



**Úplný bipartitní graf** na  $m \geq 1$  a  $n \geq 1$  vrcholech má  $m + n$  vrcholů ve dvou skupinách (partitách), přičemž hranami jsou pospojované všechny  $m \cdot n$  dvojice z různých skupin:



### Úlohy k řešení

(6.1.1) Zapišeme graf výčtem vrcholů  $\{a, b, c, d, e\}$  a zkráceným výčtem hran  $\{ac, ba, be, cd, de\}$ . Nakreslete si jej! O jaký typ grafu se jedná?



- (6.1.2) Pro jakou hodnotu  $n$  je úplný graf  $K_n$  zároveň cestou?  
 (6.1.3) Pro jakou hodnotu  $n$  je úplný graf  $K_n$  zároveň kružnicí?  
 (6.1.4) Pro jaké hodnoty  $m, n > 0$  je úplný bipartitní graf  $K_{m,n}$  zároveň kružnicí?  
 (6.1.5) Pro jaké hodnoty  $m, n > 0$  úplný bipartitní graf  $K_{m,n}$  neobsahuje žádnou kružnici?  
 (6.1.6) Kolik hran musíte přidat do kružnice délky 6, aby vznikl úplný graf na 6 vrcholech?  
 (6.1.7) Má více hran úplný graf  $K_9$  nebo úplný bipartitní graf  $K_{6,6}$ ?

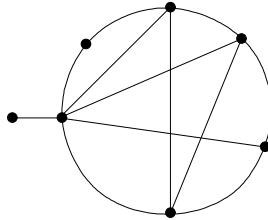
## 6.2 Stupně vrcholů v grafu

Máme-li graf, často nás zajímá, kolik z kterého vrcholu vychází hran-spojnic. Proto je jedním z prvních definovaných pojmů stupeň vrcholu v grafu.

**Definice 6.2.** *Stupněm vrcholu*  $v$  v grafu  $G$

rozumíme počet hran vycházejících z  $v$ . Stupeň značíme  $d_G(v)$ .

(Slovo “vycházející” zde nenaznačuje žádný směr; je totiž obecnou konvencí říkat, že hrana vychází z obou svých konců zároveň.)



Obrázek 6.2: Tento graf má stupně vrcholů 1, 2, 3, 4, 4, 4, 6.

**Věta 6.3.** *Součet stupňů v grafu je vždy sudý, roven dvojnásobku počtu hran.*

**Důkaz.** Při sčítání stupňů vrcholů v grafu započítáme každou hranu dvakrát – jednou za každý její konec. Proto výsledek vyjde sudý.  $\square$

### Vlastnosti stupňů

Jelikož v obyčejném grafu zvlášť nerozlišujeme mezi jednotlivými vrcholy, nemáme dáno žádné pořadí, ve kterém bychom stupně vrcholů měli psát (pokud je nepíšeme přímo do obrázku grafu). Proto si stupně obvykle seřazujeme podle velikosti.

Zajímavou otázkou je, které posloupnosti stupňů odpovídají vrcholům skutečných grafů? (Například posloupnost stupňů 1, 2, 3, 4, 5, 6, 7 nemůže nikdy nastat.)

**Věta 6.4.** *Nechť  $d_1 \leq d_2 \leq \dots \leq d_n$  je posloupnost přirozených čísel. Pak existuje graf s  $n$  vrcholy stupňů*

$$d_1, d_2, \dots, d_n$$

*právě tehdy, když existuje graf s  $n - 1$  vrcholy stupňů*

$$d_1, d_2, \dots, d_{n-d_n-1}, d_{n-d_n-1}, \dots, d_{n-2}-1, d_{n-1}-1.$$

**Komentář:** K použití Věty 6.4: Mírně nepřehledný formální zápis věty znamená, že ze seřazené posloupnosti odebereme poslední (největší) stupeň  $d_n$  a od tolika  $d_n$  bezprostředně předchozích stupňů odečteme jedničky. Zbylé stupně na začátku posloupnosti se nezmění. (Nezapomeňme, že posloupnost je třeba znovu seřadit dle velikosti.)

**Příklad 6.5.** *Existuje graf se stupni vrcholů:*

(1, 1, 1, 2, 3, 4) ?

Dle Věty 6.4 upravíme posloupnost na (1, 0, 0, 1, 2), uspořádáme ji (0, 0, 1, 1, 2), pak znovu vše zopakujeme na (0, 0, 0, 0) a takový graf už jasně existuje.

Na druhou stranu, jak takový graf sestojíme? (Obvykle není jediný, ale nás zajímá aspoň jeden z nich.) Prostě obrátíme předchozí postup, začneme z grafu se 4 vrcholy bez hran, přidáme vrchol stupně 2 spojený se dvěma z nich a další vrchol stupně 4 takto:



(1, 1, 1, 1, 2, 3, 4, 6, 7) ?

Podobně upravíme tuto posloupnost na (1, 0, 0, 0, 1, 2, 3, 5) a uspořádáme ji (0, 0, 0, 1, 1, 2, 3, 5), pak znovu upravíme na (0, 0, -1, 0, 0, 1, 2), ale to už přece nelze – stupně v grafu nejsou záporné. Proto takový graf neexistuje.  $\square$

### Úlohy k řešení

(6.2.1) Kolik hran má graf se 17 vrcholy stupňů 4?

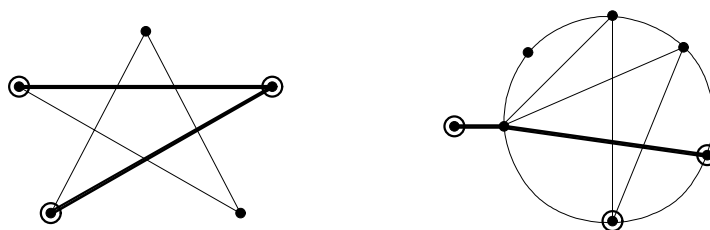
(6.2.2) Existuje graf se stupni 4, 2, 3, 5, 3, 2, 3? Nakreslete jej.

(6.2.3) Existují dva různé grafy se 6 vrcholy stupňů 2?

## 6.3 Podgrafy a Isomorfismus

Podgraf je, jednoduše řečeno, část grafu, ale toto musíme vyjádřit poněkud přesněji, abychom předešli situacím, kdy by nám zbyly hrany bez vrcholů. Navíc ještě definujeme indukovaný podgraf, který z původního grafu přebírá všechny hrany mezi vybranými vrcholy.

**Definice:** *Podgrafem* grafu  $G$  rozumíme libovolný graf  $H$  na podmnožině vrcholů  $V(H) \subseteq V(G)$ , který má za hrany libovolnou podmnožinu hran grafu  $G$  majících **oba vrcholy ve  $V(H)$** . Píšeme  $H \subseteq G$ .



Obrázek 6.3: Co je (vlevo) a co není (vpravo) podgraf.

**Definice:** *Indukovaným podgrafem* je podgraf  $H \subseteq G$  takový, který obsahuje **všechny hrany** grafu  $G$  mezi dvojicemi vrcholů z  $V(H)$ .

Pozorný čtenář si možná již při čtení předchozího textu položil otázku: Co když vezmeme jeden graf (třeba kružnici délky 5) a nakreslíme jej jednou tak, podruhé zase jinak – je to stále tentýž graf nebo ne? Přísně formálně řečeno, každé nakreslení jistého grafu, třeba té kružnice  $C_5$ , je jiným grafem, ale přitom bychom rádi řekli, že různá nakreslení téhož grafu jsou “stále stejná”. Už jen proto, že grafy mají modelovat vztahy mezi

dvojicemi objektů, ale tyto vztahy přece vůbec nezávisí na tom, jak si grafy nakreslíme. Pro tuto “stejnost” grafů se vžil pojem *isomorfní grafy*.

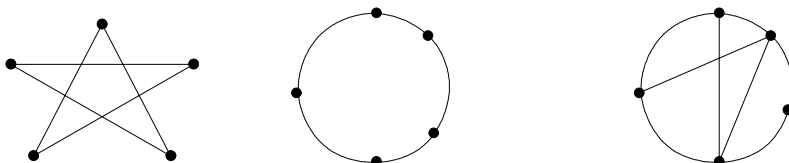
Pro správné pochopení a využití teorie grafů je tedy třeba nejprve dobře chápat pojem isomorfismu grafů.

**Definice 6.6.** *Isomorfismus* grafů  $G$  a  $H$

je bijektivní (*vzájemně jednoznačné*) zobrazení  $f : V(G) \rightarrow V(H)$ , pro které platí, že každá dvojice vrcholů  $u, v \in V(G)$  je spojena hranou v  $G$  právě tehdy, když je dvojice  $f(u), f(v)$  spojena hranou v  $H$ .

Grafy  $G$  a  $H$  jsou *isomorfní* pokud mezi nimi existuje isomorfismus. Píšeme  $G \simeq H$ .

**Fakt:** Isomorfní grafy mají stejný počet vrcholů (i hran).



**Komentář:**

Z nakreslených tří grafů jsou první dva isomorfní – jsou to jen různá nakreslení “téhož” grafu, kružnice délky 5. (Určitě snadno najdete isomorfismus mezi nimi. Pro jednoduchost isomorfismus zakreslete do obrázku tak, že vrcholy prvního očísľujete čísly 1 až 5 a vrcholy druhého stejnými čísly v pořadí odpovídajícím jeho bijekci.) Třetí graf jim isomorfní není, neboť, například, má vrcholy jiných stupňů.

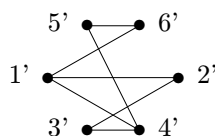
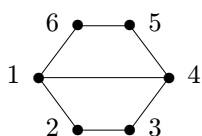
**Fakt:** Pokud je bijekce  $f$  isomorfismem, musí zobrazovat na sebe vrcholy stejných stupňů, tj.  $d_G(v) = d_H(f(v))$ . Naopak to však nestačí!

**Příklad 6.7.** Jsou následující dva grafy isomorfní?



Pokud mezi nakreslenými dvěma grafy hledáme isomorfismus, nejprve se podíváme, zda mají stejný počet vrcholů a hran. Mají. Pak se podíváme na stupně vrcholů a zjistíme, že oba mají stejnou posloupnost stupňů 2, 2, 2, 2, 3, 3. Takže ani takto jsme mezi nimi nerozlišili a mohou (nemusejí!) být isomorfní. Dále tedy nezbyvá, než zkusit všechny přípustné možnosti zobrazení isomorfismu z levého grafu do pravého.

Na levém grafu si pro ulehčení všimněme, že oba vrcholy stupně tři jsou si symetrické, proto si bez újmy na obecnosti můžeme vybrat, že nejlevější vrchol prvního grafu, označme jej 1, se zobrazí na nejlevější vrchol 1' v druhém grafu (taky stupně tři). Očísľujme zbylé vrcholy prvního grafu 2, ..., 6 v kladném smyslu, jak je ukázáno na následujícím obrázku. Druhý vrchol stupně tři, označený 4, se musí zobrazit na analogický vrchol druhého grafu (pravý spodní). Pak je již jasně vidět, že další sousedé 2, 6 vrcholu 1 se zobrazí na analogické sousedy 2', 6' vrcholu 1' v druhém grafu, a stejně je to i se zbylými vrcholy 3, 5. Výsledný isomorfismus vypadá v odpovídajícím značení vrcholů takto:



□

Abychom mohli s isomorfismem grafů přirozeně pracovat, je potřeba vědět následující fakt:

**Věta 6.8.** Relace “být isomorfní”  $\simeq$  na třídě všech grafů je ekvivalencí.

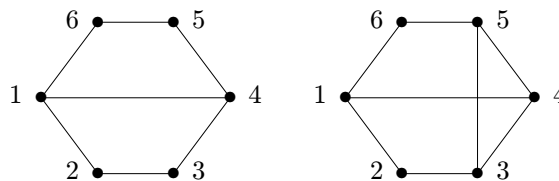
**Důkaz.** Relace  $\simeq$  je reflexivní, protože graf je isomorfní sám sobě identickým zobrazením. Relace je také symetrická, neboť bijektivní zobrazení lze jednoznačně “obrátit”. Transitivnost  $\simeq$  se snadno dokáže skládáním zobrazení–isomorfismů.  $\square$

Důsledkem je, že všechny možné grafy se rozpadnou na *třídy isomorfismu*. V praxi pak, pokud mluvíme o *grafu*, myslíme tím obvykle jeho **celou třídu isomorfismu**, tj. nezáleží nám na konkrétní prezentaci grafu.

### Další grafové pojmy

**Značení:** Mějme libovolný graf  $G$ .

- Podgrafu  $H \subseteq G$ , který je isomorfní nějaké kružnici, říkáme *kružnice v  $G$* .
- Speciálně říkáme *trojúhelník* kružnici délky 3.
- Podgrafu  $H \subseteq G$ , který je isomorfní nějaké cestě, říkáme *cesta v  $G$* .
- Podgrafu  $H \subseteq G$ , který je isomorfní nějakému úplnému grafu, říkáme *klika v  $G$* .
- Podmnožině vrcholů  $X \subseteq V(G)$ , mezi kterými nevedou v  $G$  vůbec žádné hrany, říkáme *nezávislá množina  $X$  v  $G$* .
- Indukovanému podgrafu  $H \subseteq G$ , který je isomorfní nějaké kružnici, říkáme *indukovaná kružnice v  $G$* .



**Komentář:**

První z ukázaných grafů například neobsahuje žádný trojúhelník, ale obsahuje kružnici délky 4, dokonce indukovanou. Druhý graf trojúhelník obsahuje a kružnici délky 4 také. První graf obsahuje cestu délky 4 na vrcholech 1, 2, 3, 4, 5, ale ta není indukovaná. Indukovaná cesta délky 4 v něm je třeba 2, 3, 4, 5, 6. Druhý graf tyto cesty také obsahuje, ale naopak žádná z nich není indukovaná.

První graf má největší kliku velikosti 2 – jedinou hranu, kdežto druhý graf má větší kliku na vrcholech 3, 4, 5. Největší nezávislá množina u obou grafů má 3 vrcholy 2, 4, 6.

**Příklad 6.9.** Jsou následující dva grafy isomorfní?

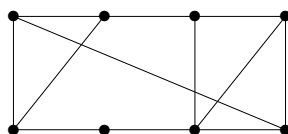


Postupovat budeme jako v Příkladě 6.7, nejprve ověříme, že oba grafy mají stejně mnoho vrcholů i stejnou posloupnost stupňů 2, 2, 2, 2, 3, 3. Pokud se však budeme snažit najít mezi nimi isomorfismus, něco stále nebude sedět, že? Co nám tedy v nalezení isomorfismu brání? Podívejme se, že v druhém grafu oba vrcholy stupně tři mají svého společného souseda, tvoří s ním trojúhelník. V prvním grafu tomu tak není, první graf dokonce nemá žádný trojúhelník. Proto zadané dva grafy nejsou isomorfní.  $\square$

**Poznámka:** Výše uvedené příklady nám ukazují některé cesty, jak poznat (tj. najít nebo vyloučit) isomorfismus dvou grafů. Ty však ne vždy musí fungovat. Čtenář se může ptát, kde tedy najde nějaký univerzální postup pro nalezení isomorfismu? Bohužel vás musíme zklamat, žádný rozumný univerzální postup není znám a má se za to, že jediná vždy fungující cesta pro nalezení či nenalezení isomorfismu mezi dvěma grafy je ve stylu *vyzkoušejte všechny možnosti* bijekcí mezi vrcholy těchto grafů. (Těch je, jak známo,  $n!$ )

### Úlohy k řešení

**(6.3.1)** Jaká je velikost největší nezávislé množiny v tomto grafu? (Tj. největší podmnožiny vrcholů, mezi kterými není žádná hrana.) Vyznačte tuto množinu.



**(6.3.2)** Jaká je délka nejkratší kružnice obsažené v grafu z Úlohy 6.3.1?

**(6.3.3)** Jaká je délka nejdelší cesty obsažené v grafu z Úlohy 6.3.1?

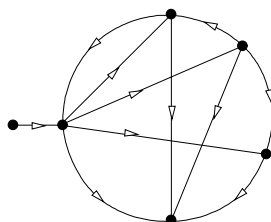
\***(6.3.4)** Jaká je délka nejdelší indukované cesty obsažené v grafu z Úlohy 6.3.1?

**(6.3.5)** Dokážete najít ke grafu z Úlohy 6.3.1 neisomorfní graf, který má stejnou posloupnost stupňů? Proč není isomorfní?

**(6.3.6)** Kolik existuje neisomorfních grafů s 5 vrcholy, všemi stupně 2?

## 6.4 Orientované grafy a multigrafy

V některých případech (například u toků v sítích) potřebujeme u každé hrany vyjádřit její směr. To vede na definici *orientovaného grafu*, ve kterém hrany jsou uspořádané dvojice vrcholů. (V obrázcích kreslíme orientované hrany se šipkami.)



**Fakt:** Orientované grafy odpovídají relacím, které nemusí být symetrické.

**Značení:** Hrana  $(u, v)$  v orientovaném grafu  $D$  začíná ve vrcholu  $u$  a končí ve (míří do) vrcholu  $v$ . Opačná hrana  $(v, u)$  je různá od  $(u, v)$ !

**Poznámka:** Navíc někdy můžeme mluvit o tzv. *multigrafech*, ve kterých padají téměř všechna omezení na hrany...

V multigrafu může mezi dvojicí vrcholů vést libovolný počet hran – tzv. *násobné hrany*, a některé z nich mohou být i orientované. Také jsou povoleny hrany, které mají oba konce totožné – tzv. *smyčky*. Multigrafy zmiňujeme jen okrajově pro úplnost, nebudeme se jimi dále zabývat.

## 6.5 Implementace grafů

Mějme jednoduchý graf  $G$  na  $n$  vrcholech a značme vrcholy jednoduše čísly  $V(G) = \{0, 1, \dots, n-1\}$ . Pro počítačovou implementaci grafu  $G$  se nabízejí dva základní způsoby:

- *Maticí sousednosti*, tj. dvourozměrným polem  $g[i][j]$ , ve kterém  $g[i][j]=1$  znamená hranu mezi vrcholy  $i$  a  $j$ .
- *Výčtem sousedů*, tj. použitím opět dvourozměrné pole  $h[i][j]$  a navíc pole  $d[i]$  stupňů vrcholů. Zde prvky  $h[i][0], h[i][1], \dots, h[i][d[i]-1]$  udávají seznam sousedů vrcholu  $i$ .

Poznámka: Dávejte si pozor na **symetrii hran** v implementaci! To znamená, že pokud uložíte hranu  $g[i][j]=1$ , tak musíte zároveň uložit i hranu  $g[j][i]=1$ , jinak se dočkáte nepříjemných překvapení. Totéž se týká i seznamů sousedů.

**Komentář:** Implementace maticí sousednosti je hezká svou jednoduchostí. Druhá možnost se však mnohem lépe hodí pro grafy s malým počtem hran, což nastává ve většině praktických aplikací. (Navíc je implementace výčtem sousedů vhodná i pro multigrafy.)

Ke grafům lze do zvláštních polí přidat také *ohodnocení vrcholů a hran* libovolnými čísly či značkami...

### Rozšiřující studium

*Rozsáhlý úvod do teorie grafů je zahrnut v [5, Kapitola 3] a navazují na něj další části této učebnice. Jiným zajímavým zdrojem, týkajícím se především implementace grafů a algoritmů na nich, mohou být výukové materiály předmětu Grafové Algoritmy na VŠB [7].*

## 6.6 Cvičení: Příklady o grafech a isomorfismu

**Příklad 6.10.** Existuje graf s posloupností stupňů  $1, 1, 3, 3, 3, 4, 4, 4, 6, 6, 6, 7$ ?

Daná posloupnost je již setříděná, jinak bychom ji nejprve vzestupně setřídili. Podle Věty 6.4 budeme posloupnost upravovat, přičemž vždy v řádku vypíšeme setříděnou posloupnost před odečtením a po odečtení (nesetříděnou).

$1, 1, 3, 3, 3, 4, 4, 4, 6, 6, 6, 7 \rightarrow 1, 1, 3, 3, 2, 3, 3, 3, 5, 5, 5$

$1, 1, 2, 3, 3, 3, 3, 3, 5, 5, 5 \rightarrow 1, 1, 2, 3, 3, 2, 2, 2, 4, 4$

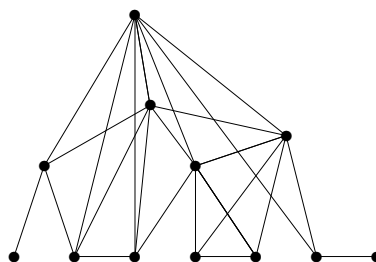
$1, 1, 2, 2, 2, 2, 3, 3, 4, 4 \rightarrow 1, 1, 2, 2, 2, 1, 2, 2, 3$

$1, 1, 1, 2, 2, 2, 2, 2, 3 \rightarrow 1, 1, 1, 2, 2, 1, 1, 1$

$1, 1, 1, 1, 1, 1, 2, 2 \rightarrow 1, 1, 1, 1, 1, 0, 1$

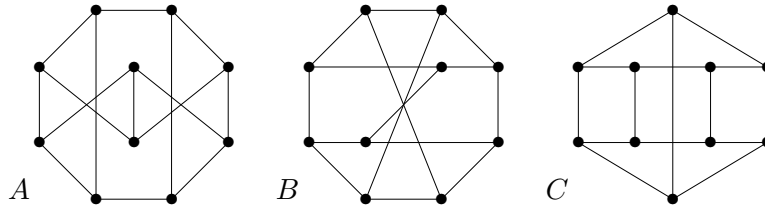
$0, 1, 1, 1, 1, 1, 1$  zřejmě tento graf existuje (3 samostatné hrany).

K poslední posloupnosti snadno sestrojíme graf, proto i graf s původní posloupností stupňů existuje. Zpětným přidáváním vrcholů sestrojíme i původní graf:



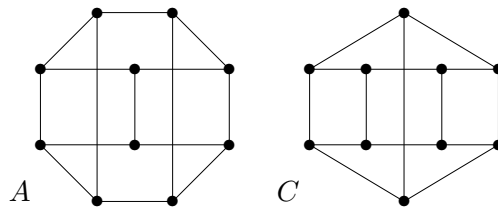
□

**Příklad 6.11.** Najděte všechny isomorfní dvojice grafů v následujících obrázcích tří 10-vrcholových grafů. Isomorfní dvojice odpovídajícím způsobem očísľujte, u neisomorfních toto zdůvodněte.



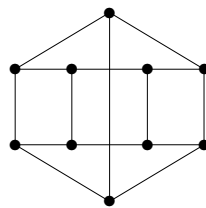
Postupujme systematicky: Všechny tři grafy mají po 10 vrcholech a všechny vrcholy stupňů 3. Takto jsme je tedy nijak nerozlišili. Podívejme se třeba na trojúhelníky v nich – opět si nepomůžeme, neboť žádný z nich trojúhelníky neobsahuje. Co se tedy podívat na obsažené kružnice  $C_4$ ? Graf  $C$  jich jasně obsahuje pět, graf  $A$  po chvíli zkoumání také, ale v grafu  $B$  najdeme i při vší snaze jen tři kružnice délky 4. (Obdobného rozdílu si můžeme všimnout, pokud se zaměříme na kružnice  $C_5$ , zkuste si to sami.)

Takže co z dosavadního zkoumání plyne? Graf  $B$  nemůže být isomorfní žádnému z  $A, C$ . Nyní tedy zbývá najít (očekávaný) isomorfismus mezi grafy  $A$  a  $C$ . To se nám skutečně podaří poměrně snadno - stačí “prohozením” prostředních dvou vrcholů u grafu  $A$  získat lepší obrázek



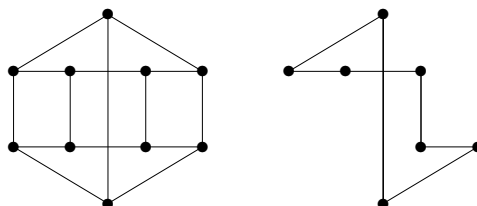
a odpovídající bijekce je na pohled zřejmá. (Doplňte si ji společným očíslováním vrcholů obou grafů.)  $\square$

**Příklad 6.12.** Jaká je nejdelší kružnice a nejdelší indukovaná kružnice obsažená v následujícím grafu? Zdůvodněte.



Co se týče nejdelší kružnice, ta nemůže být z principu delší než počet všech vrcholů, tedy 10. Přitom kružnici délky 10 v zadaném grafu snadno najdeme.

Co se týče indukované kružnice, ta musí s vybranými vrcholy podgrafu obsahovat i všechny hrany mezi nimi (a přesto to musí stále být jen kružnice). To zřejmě žádnou kružnici délky 10 splněno není. Nenajdeme dokonce ani takové kružnice délky 9 a 8, nejdelší při troše snahy najdeme indukovanou kružnici délky 7, jako třeba na následujícím obrázku:



Závěrem se zamysleme, proč vlastně delší indukovanou kružnici (než 7) nelze v našem grafu nalézt. Pokud bychom, pro spor, našli indukovanou kružnici délky 8, tak by musela použít jak ze spodního, tak z horního pětiúhelníku po čtyřech vrcholech. (Nemůže totiž být všech 5 vrcholů z jednoho pětiúhelníku, neboť to by už byla kružnice délky jen 5.) Sami však snadným pokusem zjistíme, že v takové kružnici budou ještě hrany navíc, tudíž nebude indukovaná.  $\square$

### Úlohy k řešení

(6.6.1) Existuje graf s posloupností stupňů 1, 1, 1, 3, 3, 3, 4, 4, 4?

Návod: Použijte Větu 6.4.

(6.6.2) Existuje graf s posloupností stupňů 1, 1, 1, 1, 1, 1, 1, 1, 6, 6?

(6.6.3) Kolik existuje neisomorfních grafů s 6 vrcholy, všemi stupně 3? Nakreslete je.

Návod: Podívejte se místo těchto grafů na jejich doplňky – dejte hrany právě tam, kde je původní graf nemá. Tím vzniknou grafy se všemi stupni  $5 - 3 = 2$ .

(6.6.4) Kolik hran má graf s touto posloupností stupňů?

A: 1, 1, 2, 2, 3, 3, 4, 4, 4, 4, 5, 6, 7

B: 1, 1, 2, 2, 2, 2, 4, 4, 4, 5, 6, 7

C: 1, 1, 2, 2, 2, 3, 3, 4, 4, 4, 5, 6, 7

(6.6.5) Najděte a vyznačte isomorfismus mezi následujícími dvěma grafy na 7 vrcholech.



(6.6.6) Vyznačte isomorfismus mezi následujícími dvěma grafy na 8 vrcholech:



(6.6.7) Vyznačte isomorfismus mezi následujícími dvěma grafy na 8 vrcholech:



\*(6.6.8) Existují dva neisomorfní grafy s posloupnostmi stupňů

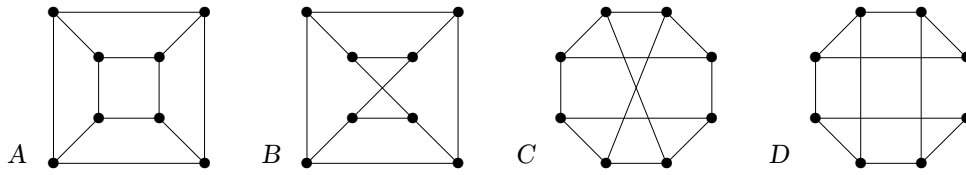
A: 3, 3, 3, 3, 3, 3,

B: 2, 2, 3, 3 ?

U možnosti, kde dva neisomorfní grafy existují, je nakreslete. Pokud neexistují, pokuste se to správně zdůvodnit. (Pamatujte, že uvažujeme jednoduché grafy, tj. grafy bez smyček a násobných hran. Grafy nemusí být souvislé.)

\*(6.6.9) Najděte mezi následujícími grafy všechny isomorfní dvojice a zdůvodněte svou odpověď. (Isomorfní dvojice stejně očísľujte, u neisomorfních nejděte odlišnosti.)





Návod: Pokud vám třeba jedno očíslování pro isomorfismus nevyjde, musíte zkoušet další a další a probrat tak všechny možnosti.

**(6.6.10)** Jaká je nejdelší kružnice a nejdelší indukovaná kružnice obsažená v grafech A a B z Úlohy 6.6.9?

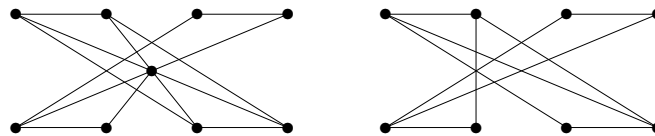
\***(6.6.11)** Kolik podgrafů úplného grafu  $K_{10}$  je isomorfních kružnici  $C_4$ ?

\***(6.6.12)** Najděte všechny neisomorfní grafy se stupni 3, 3, 3, 3, 2, 2, 2.

Návod: Uvědomte si, že vrcholy stupně 2 lze “zanedbat” – nahradit hranami. Ve výsledku pak zbudou 4 vrcholy stupňů 3, ale mezi nimi mohou být násobné hrany nebo i smyčky. Kreslete si obrázky.

**(6.6.13)** Vraťte se zpět k příkladům a úlohám z Oddílu 6.3 a zkuste, zda je se znalostí nových metod jste schopni řešit lépe a elegantněji.

**(6.6.14)** Kolik nejvíce vrcholů obsahuje nezávislá množina v nakreslených grafech? Tuto největší nezávislou množinu si v grafech vyznačte.

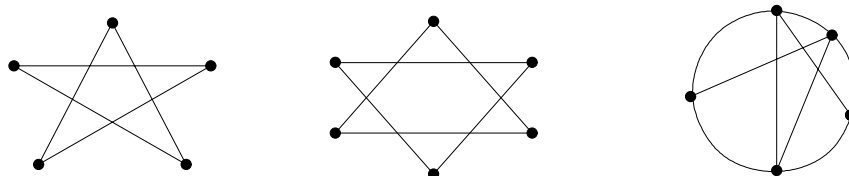


## 7 Souvislost grafu

### Úvod

Pokud máme graf, který modeluje nějaká spojení či síť, přirozeně nás zajímá, jakou máme možnost se dostat odněkud někam v tomto grafu. To má množství praktických motivací – například počítačové, dopravní, telefonní či potrubní sítě. Je pochopitelné, že v takových sítích chceme mít možnost se dostat z každého místa do každého jiného.

Grafům s takovou vlastností říkáme souvislé. (Abychom si ujasnili terminologii, když mluvíme o souvislosti, nejedná se o žádné “hledání souvislosti grafů s něčím jiným”, ale o možnost procházení mezi vrcholy jednoho grafu po jeho hranách.) Pro ukázkou se podívejme na následující obrázky tří grafů:



Poznáte, který z nich je nesouvislý? Všechny tři vypadají “spojeně”, ale podívejte se důkladně na ten prostřední – v něm není žádné propojení mezi vrchním a spodním vrcholem po hranách (křížení se nepočítají).

Další potřebnou znalostí je umět souvislý graf algoritmicky celý procházet. Zde si uvedeme obecnou kostru algoritmu pro procházení grafu a zmíníme některé konkrétní variace tohoto algoritmu. (Znáte algoritmus pro procházení bludiště? V grafech je to v obecnosti podobné.) Také se později zmíníme i o vyšších stupních souvislosti grafu – o zálohování spojení v grafech pro případy výpadku.

## Cíle

Tato lekce definuje pojmy souvislosti a komponent grafu. Cílem je ukázat, jak se souvislostí grafu pracovat a jak algoritmicky graf procházet po jeho hranách. Zmíněny jsou i vyšší stupně souvislosti.

## 7.1 Spojení vrcholů, komponenty

Nejprve bychom si měli přesně ujasnit, jak se pohybujeme grafem, tedy co je vlastně procházkou v grafu. Tento pojem by měl postihnout základní věc, že v grafu procházíme hranami vždy z vrcholu do sousedního vrcholu, a přitom ponechat dostatek volnosti pro vrácení se a zacyklení procházek.

**Definice:** *Sledem* délky  $n$  v grafu  $G$  rozumíme posloupnost vrcholů a hran

$$v_0, e_1, v_1, e_2, v_2, \dots, e_n, v_n,$$

ve které vždy hrana  $e_i$  má koncové vrcholy  $v_{i-1}, v_i$ .

**Komentář:** Sled je vlastně procházka po hranách grafu z  $u$  do  $v$ . Příkladem sledu může být průchod IP paketu internetem (včetně cyklení).

**Lema 7.1.** *Mějme relaci  $\sim$  na množině vrcholů  $V(G)$  libovolného grafu  $G$  takovou, že pro dva vrcholy  $u \sim v$  právě když existuje v  $G$  sled začínající v  $u$  a končící ve  $v$ . Pak  $\sim$  je relací ekvivalence.*

**Důkaz.** Relace  $\sim$  je reflexivní, neboť každý vrchol je spojený sám se sebou sledem délky 0. Symetrická je také, protože sled z  $u$  do  $v$  snadno obrátíme na sled z  $v$  do  $u$ . Stejně tak je  $\sim$  tranzitivní, protože dva sledy můžeme na sebe navázat v jeden.  $\square$

**Definice:** Třídy ekvivalence výše popsané (Lema 7.1) relace  $\sim$  na  $V(G)$  se nazývají *komponenty souvislosti* grafu  $G$ .

Jinak se taky *komponentami souvislosti myslí podgrafy* indukované na těchto třídách ekvivalence.

Připomeňme si, že cesta je vlastně sledem bez opakování vrcholů.

**Věta 7.2.** *Pokud mezi dvěma vrcholy grafu  $G$  existuje sled, pak mezi nimi existuje cesta.*

**Důkaz.** Necht'  $u = v_0, e_1, v_1, \dots, e_n, v_n = v$  je sled délky  $n$  mezi  $u$  a  $v$  v  $G$ . Začneme budovat nový sled  $W$  z vrcholu  $w_0 = u$ , který už bude cestou:

Předpokládejme, že nový sled  $W$  už má počátek  $w_0, e_1, w_1, \dots, w_i$  (na začátku  $i = 0$ , tj. jen  $w_0$  bez hran), kde  $w_i = v_j$  pro některé  $j \in \{0, 1, \dots, n\}$ . Najdeme největší index  $k \geq j$  takový, že  $v_k = v_j = w_i$ , a sled  $W$  pokračujeme krokem  $\dots, w_i = v_j = v_k, e_{k+1}, v_{k+1} = w_{i+1}$ . Zbývá dokázat, že nový vrchol  $w_{i+1} = v_{k+1}$  se ve sledu  $W$  neopakuje. Pokud by tomu ale tak bylo  $w_l = w_{i+1}$ ,  $l \leq i$ , pak bychom na vrchol  $w_{i+1}$  "přeskočili" už dříve z vrcholu  $w_l$ , spor. Nakonec skončíme, když  $w_i = v$ .  $\square$

**Komentář:** Ačkoliv uvedený důkaz vypadá složitě, je to jen jeho formálním zápisem. Ve skutečnosti se v důkaze neděje nic jiného, než že se původní sled zkracuje o opakované vrcholy, až nakonec zákonitě vznikne cesta.

Závěrem se dostáváme k nejdůležitější definici souvislého grafu:

**Definice 7.3.** *Graf  $G$  je souvislý*

pokud je  $G$  tvořený jedinou komponentou souvislosti, tj. pokud každé dva vrcholy  $G$  jsou spojené cestou (dle Věty 7.2).

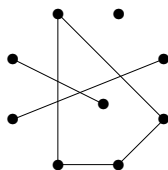
Poznámka: Prázdný graf je souvislý.

### Úlohy k řešení

(7.1.1) Který z těchto dvou grafů je souvislý?



(7.1.2) Kolik komponent souvislosti má tento graf?



## 7.2 Prohledávání grafu

Když se lidé dívají na grafy, obvykle je vnímají jako obrázky a jejich pohled je jakoby globální. Pokud je však graf zpracováván v počítači (a obzvláště pokud se jedná o skutečně velký graf), tento globální pohled nemáme k dispozici a graf musíme *prohledávat lokálně*. Z toho důvodu se potřebujeme seznámit, jako vůbec s prvním grafovým algoritmem, s obecným postupem *lokálního prohledávání grafu*. Tento algoritmus není složitý a víceméně zobecňuje dobře známý postup procházení všech cest v bludišti.

Pro vytvoření co nejobecnějšího schématu algoritmu pro procházení grafu vystačíme s následujícími datovými stavy a pomocnou strukturou:

- **Vrchol:** má stavy ...
  - iniciační – dostane na začátku,
  - nalezený – poté, co jsme jej přes některou hranu našli,
  - zpracovaný – poté, co jsme už probrali všechny hrany z něj vycházející.
- **Hrana:** má stavy ...
  - iniciační – dostane na začátku,
  - zpracovaná – poté, co už byla probrána od jednoho ze svých vrcholů.
- **Úschovna:** je pomocná datová struktura (množina),
  - udržuje nalezené a ještě nezpracované vrcholy.

Poznámka: Způsob, kterým se vybírají vrcholy z úschovny ke zpracování, určuje variantu algoritmu procházení grafu. V prohledávaných vrcholech a hranách se pak provádějí konkrétní programové akce pro prohledání a zpracování našeho grafu.

### Algoritmus 7.4. *Procházení souvislých komponent grafu*

Algoritmus projde a zpracuje každou hranu a vrchol grafu  $G$ . Průchod se děje po jednotlivých komponentách, po projití jedné komponenty se přejde na případnou další. Konkrétní programové akce potřebné ke zpracování grafu jsou uvedeny v pomocných funkcích ZPRACUJ().

```
vstup < graf G;
```

```

stav(všechny vrcholy a hrany G) = iniciační;
uschovna U = {libovolný vrchol  $v_0$  grafu G};
stav( $v_0$ ) = nalezený;
// zpracování vybrané komponenty G
while (U je neprázdná) {
    vybrat  $v \in U$ ;     $U = U \setminus \{v\}$ ;
    ZPRACUJ( $v$ );
    for (e hrany vycházející z  $v$ ) {
        if (stav(e) == iniciační) ZPRACUJ(e);
         $w =$  druhý vrchol  $e = vw$ ;
        if (stav( $w$ ) == iniciační) {
            stav( $w$ ) = nalezený;
             $U = U \cup \{w\}$ ;
        }
        stav(e) = zpracovaná;
    }
    stav( $v$ ) = zpracovaný;
// případný přechod na další komponentu G
if (U je prázdná && G má další komponenty)
    uschovna U = {vrchol  $v_1$  další komponenty G};
}

```

Příklady aplikací schématu Algoritmu 7.4 jsou uvedeny dále například v Důsledku 8.4 a v Algoritmu 8.9. Konkrétní postup algoritmu procházení bude ilustrován ve Cvičení 7.4.

### Způsoby implementace procházení grafu

- *Procházení “do hloubky”* – úschovna  $U$  je implementovaná jako zásobník, tj. dále prohledáváme od posledních nalezených vrcholů.
- *Procházení “do šířky”* – úschovna  $U$  je implementovaná jako fronta, tj. dále prohledáváme od prvních nalezených vrcholů.
- *Dijkstrův algoritmus* pro nejkratší cestu – z úschovny vybíráme vždy vrchol nejbližší k počátečnímu  $v_0$ . (Toto je dost podobné prohledávání do šířky, ale obecnější i pro případy, kdy hrany nejsou “stejně dlouhé”.)  
Tento algoritmus bude v příští přednášce.

### 7.3 Vyšší stupně souvislosti

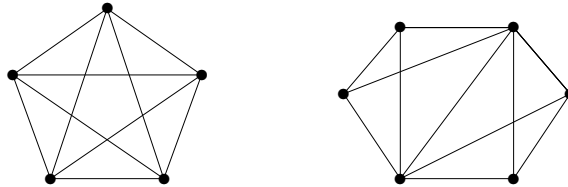
V síťových aplikacích nás často zajímá nejen, jestli se za normálních podmínek můžeme pohybovat mezi vrcholy/uzly, ale také, jaké spojení můžeme nalézt v případě lokálních výpadků (odolnost a redundance). Toto lze teoreticky podchytit zkoumáním “vyšších” stupňů souvislosti grafu.

**Definice:** Graf  $G$  je *hranově  $k$ -souvislý*,  $k > 1$ , pokud i po odebrání libovolných  $k - 1$  hran z  $G$  zůstane výsledný graf souvislý.

**Definice:** Graf  $G$  je *vrcholově  $k$ -souvislý*,  $k > 1$ , pokud i po odebrání libovolných  $k - 1$  vrcholů z  $G$  zůstane výsledný graf souvislý.

Speciálně úplný graf  $K_n$  je vrcholově  $(n - 1)$ -souvislý.

**Komentář:** Stručně řečeno, vysoká hranová souvislost znamená vysoký stupeň odolnosti sítě proti výpadkům spojení-hran, neboli síť zůstane stále dosažitelná, i když libovolných  $k - 1$  spojení bude přerušeno. Vysoká vrcholová souvislost je mnohem silnějším pojmem, znamená totiž, že síť zůstane dosažitelná i po výpadku libovolných  $k - 1$  uzlů-vrcholů (samozřejmě mimo těch vypadlých uzlů).



Na ilustračním obrázku má první graf vrcholovou souvislost 4 a snadno vidíme, že po odebrání tří vrcholů či hran zůstává souvislý. Z druhého grafu bychom museli odebrat nejméně 3 hrany, aby se stal nesouvislým, a proto je jeho hranová souvislost 3. Na druhou stranu však stačí odebrat 2 vrcholy, aby mezi jeho levým a pravým krajním vrcholem žádné spojení nezůstalo. (Vidíte, které dva?)

Důkaz následujícího výsledku (Mengerova věta) je příliš obtížný pro náš předmět, ale výsledek sám určitě stojí za zapamatování a pozdější použití:

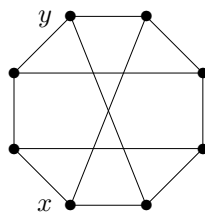
**Věta 7.5.** *Graf  $G$  je hranově  $k$ -souvislý právě když mezi libovolnými dvěma vrcholy lze vést aspoň  $k$  hranově-disjunktních cest (vrcholy mohou být sdílené). Graf  $G$  je vrcholově  $k$ -souvislý právě když mezi libovolnými dvěma vrcholy lze vést aspoň  $k$  disjunktních cest (různých až na ty dva spojované vrcholy).*

**Komentář:** Věta nám vlastně říká, že stupeň souvislosti grafu se přirozeně rovná stupni redundance spojení vrcholů. Na výše uvedeném obrázku mezi každými dvěma vrcholy prvního grafu můžeme vést až 4 disjunktní cesty. U druhého grafu třeba mezi levým a pravým koncem lze vést jen 2 (vrcholově) disjunktní cesty, ale mezi každými dvěma vrcholy lze vést 3 hranově-disjunktní cesty.

### Úlohy k řešení

(7.3.1) *Jaký stupeň souvislosti má úplný bipartitní graf  $K_{n,n}$ ?*

(7.3.2) *Kolik nejméně vrcholů musíte vypustit z nakresleného grafu, aby v něm nezbyla žádná cesta mezi vrcholy  $x$  a  $y$ ? Zdůvodněte.*



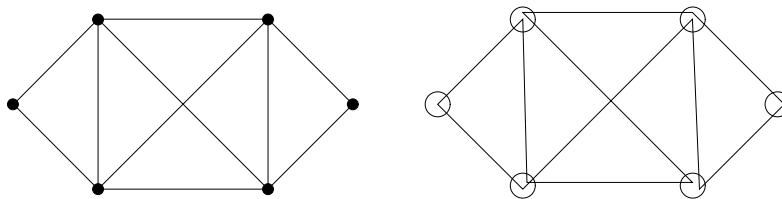
(7.3.3) *Kolik nejméně hran musíte přidat k cestě délky 7, aby vznikl vrcholově 2-souvislý graf?*

## 7.4 Jedním tahem: Eulerovské grafy

Především z historických důvodů zařazujeme na závěr tuto poznámku o kreslení grafů jedním tahem.

**Definice:** *Tah* je sled v grafu bez opakování hran.

*Uzavřený tah* je tahem, který končí ve vrcholu, ve kterém začal.

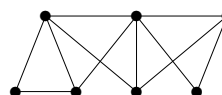


Nejstarší výsledek teorie grafů vůbec pochází od Leonarda Eulera: (Jedná se o problém slavných 7 mostů v Královci / Königsbergu / dnešním Kaliningradě.)

**Věta 7.6.** Graf  $G$  lze nakreslit jedním uzavřeným tahem právě když  $G$  je souvislý a všechny vrcholy v  $G$  jsou sudého stupně.

**Důsledek 7.7.** Graf  $G$  lze nakreslit jedním otevřeným tahem právě když  $G$  je souvislý a všechny vrcholy v  $G$  až na dva jsou sudého stupně.

### Úlohy k řešení



(7.4.1) Lze tento graf nakreslit jedním otevřeným tahem?

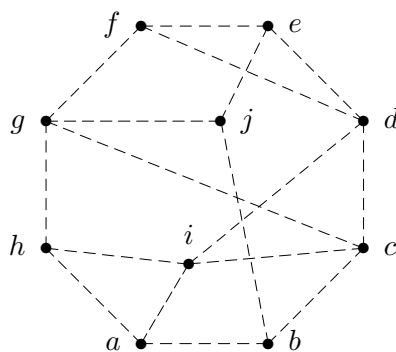
(7.4.2) Kolik hran musíte přidat ke grafu z předchozí otázky, aby se dal nakreslit jedním uzavřeným tahem?

### Rozšiřující studium

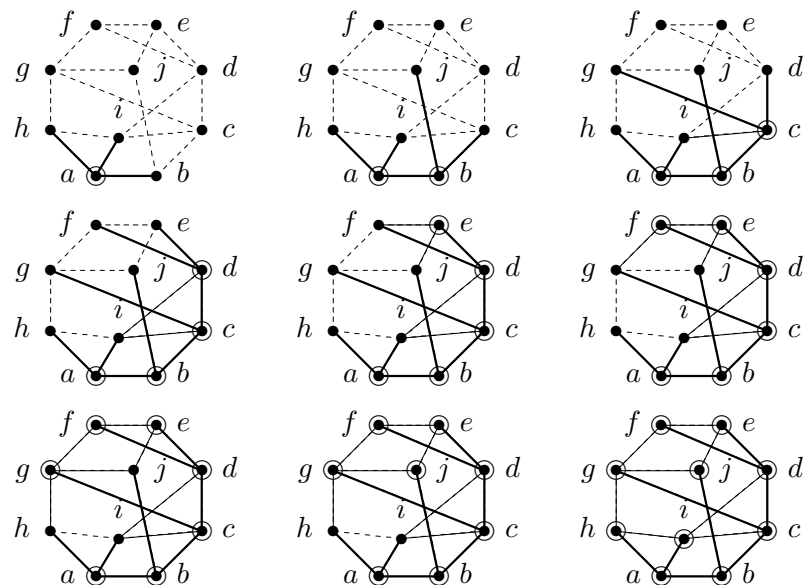
Souvislosti grafů se blíže teoreticky věnují [5, Oddíly 3.2,8]. Algoritmy pro procházení grafu jsou podrobně popsány (včetně netriviálních aplikací) v [3] a demonstrovány i v [4].

## 7.5 Cvičení: Příklady na souvislost grafů

**Příklad 7.8.** Ukázka průchodu následujícím grafem do hloubky z vrcholu  $a$ .

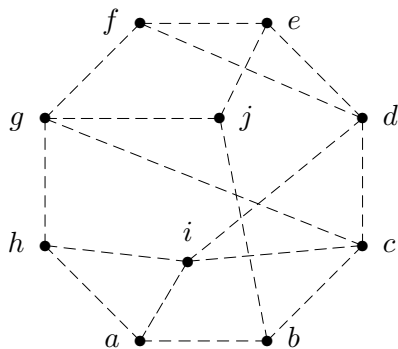


V této ukázce budeme obrázky znázorňovat stavy Algoritmu 7.4 v jednotlivých krocích takto: Neprohledané hrany jsou čárkované, prohledané hrany plnou čarou a hrany, které vedly k nalezení vrcholů, jsou tlustou čarou (tyto hrany často mívají speciální význam v aplikacích schématu algoritmu). Nalezené vrcholy se poznají podle příchozí tlusté hrany a zpracované vrcholy jsou značené dvojím kroužkem.

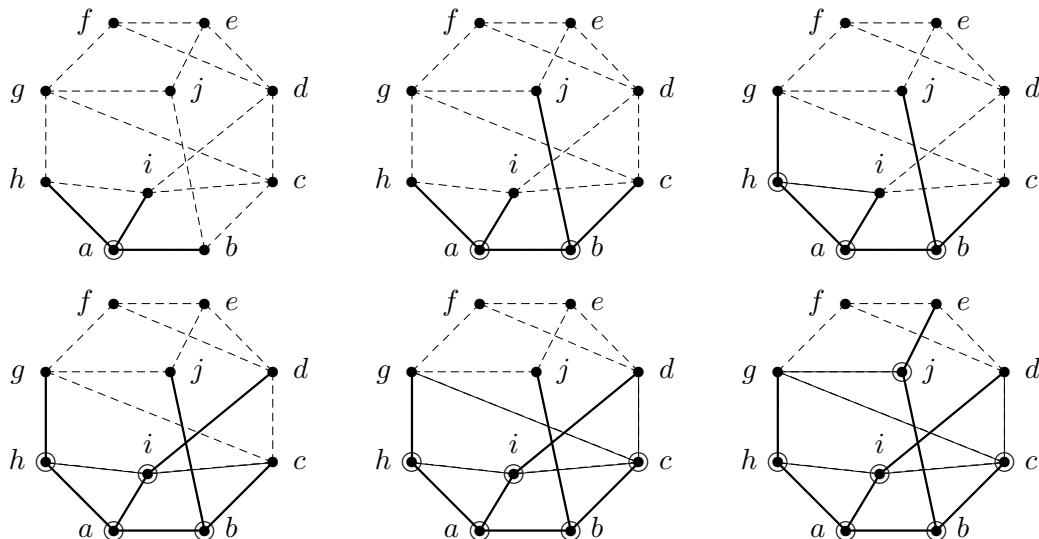


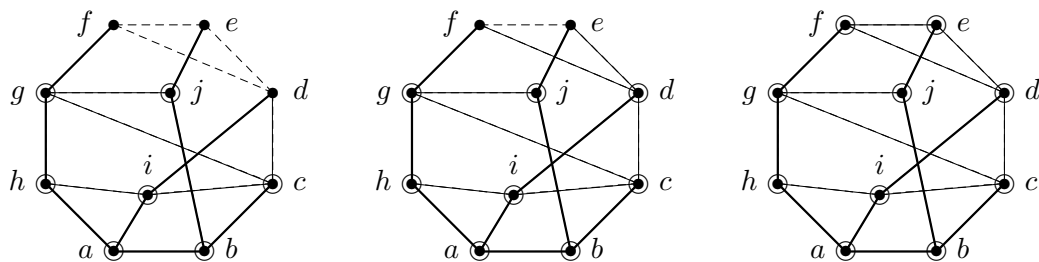
Tímto zpracování zadaného grafu skončilo. Mimo jiné jsme zjistili, že graf má jedinou komponentu souvislosti.  $\square$

**Příklad 7.9.** Ukázka průchodu následujícím grafem do šířky z vrcholu  $a$ .



V této ukázce budeme obrázky znázorňovat stavy Algoritmu 7.4 stejně jako v předchozím příkladě.





Tímto zpracování zadaného grafu skončilo. Vidíte rozdíly tohoto průchodu proti předchozímu příkladu?  $\square$

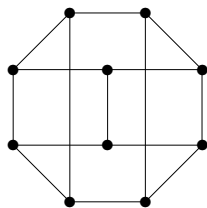
**Příklad 7.10.** Mějme graf  $H_3$ , jehož vrcholy jsou všechny podmnožiny množiny  $\{1, 2, 3\}$  a hrany spojují právě disjunktní dvojice podmnožin. (Tj.  $H_3$  má 8 vrcholů.) Rozhodněte, zda je  $H_3$  souvislý graf, a napište, kolik má  $H_3$  hran.

Vrchol  $\emptyset$  odpovídající prázdné množině je spojený se všemi sedmi ostatními vrcholy, takže je graf souvislý. Nakreslete si jej!

Každý vrchol  $i$ -prvkové podmnožiny je pak spojený s  $2^{3-i}$  disjunktními podmnožinami doplňku. Celkem tedy máme součtem všech stupňů  $\frac{1}{2}(7 + 3 \cdot 2^2 + 3 \cdot 2^1 + 2^0) = 13$  hran.  $\square$

### Úlohy k řešení

(7.5.1) Kolik komponent souvislosti má tento graf?



(7.5.2) Kolik nejvýše komponent může mít graf s 15 vrcholy, všemi stupně 2?

\*(7.5.3) Kolik nejvýše komponent může mít graf s 30 vrcholy, všemi stupně 4?

(7.5.4) Kolik existuje neisomorfních grafů s 9 vrcholy, všemi stupně 2? Nakreslete je všechny a nezapomeňte na ty nesouvislé.

(7.5.5) Mějme graf  $H_5$ , jehož vrcholy jsou všechny dvouprvkové podmnožiny množiny  $\{1, 2, 3, 4, 5\}$  a hrany spojují právě disjunktní dvojice podmnožin. (Tj.  $H_5$  má 10 vrcholů.) Rozhodněte, zda je  $H_5$  souvislý graf, a napište, kolik má  $H_5$  hran.

(7.5.6) Kolik nejméně hran musí mít graf na 12 vrcholech, aby stupeň jeho souvislosti byl 3?

(7.5.7) Kolik nejvíce hran může mít graf na 10 vrcholech,

a) který se skládá ze tří komponent souvislosti,

b) jehož každá komponenta souvislosti má nejvíce tři vrcholy?

Tento graf také nakreslete.

\*(7.5.8) Existují dva neisomorfní grafy se stejnou posloupností stupňů

A: 2,2,2,3,3,

B: 2,3,3,3,3 ?

U možnosti, kde dva neisomorfní grafy existují, je nakreslete. U zbylé možnosti pak správně zdůvodněte, proč dva takové neisomorfní grafy neexistují. (Pamatujte, že uvažujeme jednoduché grafy, tj. grafy bez smyček a násobných hran. Grafy nemusí být souvislé.)

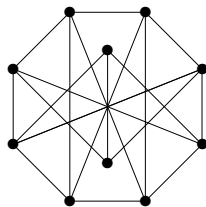


- \*(7.5.9) Existují dva neisomorfní grafy se stejnou posloupností stupňů  
 A: 2,2,2,1,1,  
 B: 2,3,3,3,3 ?
- \*(7.5.10) Existují dva neisomorfní grafy se stejnou posloupností stupňů  
 A: 2,2,2,2,2,  
 B: 1,1,3,3,3,3 ?
- \*(7.5.11) Kolik existuje neisomorfních grafů na 6 vrcholech s posloupností stupňů  
 1, 1, 2, 2, 2, 2 ?

Všechny tyto grafy nakreslete a zdůvodněte i, proč jich není více. (Pamatujte, že uvažujeme jednoduché grafy, tj. grafy bez smyček a násobných hran. Grafy nemusí být souvislé.)

- \*(7.5.12) Hamiltonovská kružnice v grafu  $G$  je takový podgraf, který je isomorfní kružnici a obsahuje všechny vrcholy  $G$ . (Neboli je to kružnice procházející všemi vrcholy  $G$  právě jednou.) Najděte a nakreslete jednoduchý neorientovaný graf, který zároveň je vrcholově 3-souvislý a přitom neobsahuje Hamiltonovskou kružnici.

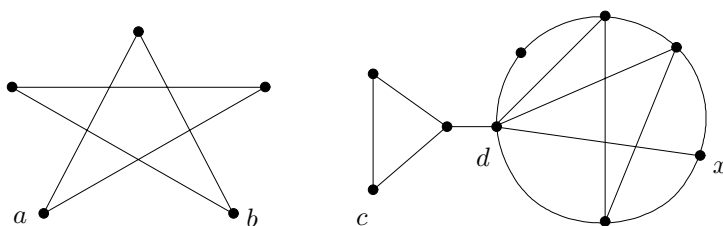
- (7.5.13) Lze tento graf nakreslit jedním tahem? A uzavřeným? (Uprostřed není vrchol, jen se tam kříží hrany.)



## 8 Vzdálenost a metrika

### Úvod

V minulé lekci jsme mluvili o souvislosti grafu, tj. o možnosti procházení z jednoho vrcholu do jiného. Někdy je prostá informace o souvislosti dostačující, ale většinou bychom rádi věděli i jak je to z jednoho vrcholu do druhého “daleko”. Proto se nyní podíváme, jak krátká či dlouhá taková procházka mezi dvěma vrcholy grafu je.



V jednodušším případě se při zjišťování grafové vzdálenosti díváme jen na minimální počet prošlých hran z vrcholu do vrcholu. Tak například v našem ilustračním obrázku je vzdálenost mezi  $a$ ,  $b$  rovna 2, vzdálenost mezi  $a$ ,  $c$  je rovna  $\infty$  a vzdálenost mezi  $c$ ,  $x$  je rovna 3. Navíc vidíme, že vrchol  $d$  má “centrální” pozici v pravém grafu – každý další vrchol je od něj ve vzdálenosti nejvýše 2, kdežto vrchol  $c$  má “okrajovou” pozici.

V obecném případě však při určování vzdálenosti bereme do úvahy délky jednotlivých hran podél cesty (tyto délky musí být nezáporné!). Jako hlavní náplň lekce si uvedeme známý Dijkstrův algoritmus pro určování nejkratší cesty v grafu. (Tento algoritmus je, mimo jiné aplikace, také základem programů vyhledávajících vlaková/autobusová spojení.) Většina tvrzení a algoritmy obsažené v této přednášce je beze změny aplikovatelná i na orientované grafy (tj. když nás zajímá i směr procházení hran).

## Cíle

Prvním cílem této lekce je definovat vzdálenost v grafech a její základní vlastnosti. Dalším úkolem je ukázat a správně pochopit Dijkstrův algoritmus pro hledání nejkratší cesty v grafu.

## 8.1 Vzdálenost v grafu

Vzpomeňme si, že sledem délky  $n$  v grafu  $G$  rozumíme posloupnost vrcholů a hran  $v_0, e_1, v_1, e_2, v_2, \dots, e_n, v_n$ , ve které hrana  $e_i$  má koncové vrcholy  $v_{i-1}, v_i$ .

**Definice 8.1.** *Vzdálenost*  $d_G(u, v)$  dvou vrcholů  $u, v$  v grafu  $G$  je dána délkou nejkratšího sledu mezi  $u$  a  $v$  v  $G$ . Pokud sled mezi  $u, v$  neexistuje, je vzdálenost  $d_G(u, v) = \infty$ .

**Komentář:** Neformálně řečeno, vzdálenost mezi  $u, v$  je rovna *nejmenšímu počtu hran*, které musíme projít, pokud se chceme dostat z  $u$  do  $v$ . Speciálně  $d_G(u, u) = 0$ . Uvědomme si, že nejkratší sled je vždy cestou (vrcholy se neopakují) – Věta 7.2.

Grafová vzdálenost se chová dosti podobně běžné vzdálenosti, jak ji známe z geometrie, což bude vidět z následujících tvrzení.

**Poznámka:** V neorientovaném grafu je vzdálenost symetrická  $d_G(u, v) = d_G(v, u)$ .

**Lema 8.2.** *Vzdálenost v grafech splňuje trojúhelníkovou nerovnost:*

$$\forall u, v, w \in V(G) : d_G(u, v) + d_G(v, w) \geq d_G(u, w).$$

**Důkaz.** Nerovnost snadno plyne ze zřejmého pozorování, že na sled délky  $d_G(u, v)$  mezi  $u, v$  lze navázat sled délky  $d_G(v, w)$  mezi  $v, w$ , čímž vznikne sled délky  $d_G(u, v) + d_G(v, w)$  mezi  $u, w$ . Skutečná vzdálenost mezi  $u, w$  pak už může být jen menší.  $\square$

**Věta 8.3.** *Nechť  $u, v, w$  jsou vrcholy souvislého grafu  $G$  takové, že  $d_G(u, v) < d_G(u, w)$ . Pak při algoritmu procházení grafu  $G$  do šířky z vrcholu  $u$  je vrchol  $v$  nalezen dříve než vrchol  $w$ .*

**Důkaz.** Postupujeme indukcí podle vzdálenosti  $d_G(u, v)$ : Pro  $d_G(u, v) = 0$ , tj.  $u = v$  je tvrzení jasné – vrchol  $u$  jako počátek prohledávání byl nalezen první. Proto nechť  $d_G(u, v) = d > 0$  a označme  $v'$  souseda vrcholu  $v$  bližšího k  $u$ , tedy  $d_G(u, v') = d - 1$ . Stejně tak značme  $w'$  souseda vrcholu  $w$  bližšího k  $u$ , tedy  $d_G(u, w') < d_G(u, w)$ . Potom byl vrchol  $v'$  nalezen v prohledávání do šířky dříve než vrchol  $w'$  podle indukčního předpokladu. To znamená, že  $v'$  se dostal do fronty úschovny dříve než  $w'$ , a tudíž sousedé  $v'$  (mezi nimi  $v$ ) budou při prohledávání nalezeni dříve než sousedé  $w'$ .  $\square$

**Důsledek 8.4.** *Základní algoritmus procházení grafu do šířky lze použít pro výpočet vzdáleností z vrcholu  $u$  do ostatních vrcholů grafu:*

Toto je poměrně jednoduchá aplikace, kdy počátečnímu vrcholu přiřadíme vzdálenost 0, a pak vždy každému dalšímu nalezenému vrcholu přiřadíme vzdálenost o 1 větší než byla vzdálenost vrcholu, ze kterého jsme jej našli.

**Komentář:** My si ale později ukážeme obecnější Dijkstrův algoritmus, který počítá nejkratší vzdálenost při libovolně kladně ohodnocených délkách hran.

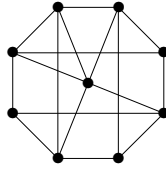
## Úlohy k řešení

(8.1.1) *Jaká je největší vzdálenost dvou vrcholů v úplném grafu?*

(8.1.2) *Jaká je největší vzdálenost dvou vrcholů v úplném bipartitním grafu  $K_{33,44}$ ?*

(8.1.3) Jaká je největší vzdálenost dvou vrcholů v kružnici  $C_{11}$ ?

(8.1.4) Jaká je největší vzdálenost mezi dvěma vrcholy v následujícím grafu?



## 8.2 Výpočet metriky

Než se podíváme na samotný postup výpočtu vzdálenosti z jednoho vrcholu do druhého, uvedeme si ještě “globální” pohled na soubor všech vzdáleností v grafu, tedy na tzv. metriku grafu. Zajímavostí tohoto pohledu je především to, že výpočet celé metriky najednou má velice jednoduchý algoritmus, který není příliš známý a přitom je dobře použitelný i v jiných oblastech. (Jak třeba později uvidíte v teorii automatů.)

**Značení:** Metrikou grafu myslíme soubor vzdáleností mezi všemi dvojicemi vrcholů grafu. Jinak řečeno, *metrikou grafu  $G$  je matice* (dvourozměrné pole)  $d[] []$ , ve kterém prvek  $d[i][j]$  udává vzdálenost mezi vrcholy  $i$  a  $j$ .

### Metoda 8.5. *Iterativní výpočet metriky skládáním cest*

- Na počátku nechť  $d[i][j]$  udává 1 (případně délku hrany  $\{i, j\}$ ), nebo  $\infty$  pokud hrana mezi  $i, j$  není.
- Po každém kroku  $t \geq 0$  nechť  $d[i][j]$  udává délku nejkratší cesty mezi  $i, j$ , která jde pouze přes vnitřní vrcholy z množiny  $\{0, 1, 2, \dots, t-1\}$ .
- Při přechodu z  $t$  na následující krok  $t+1$  upravujeme vzdálenost pro každou dvojici vrcholů – jsou vždy pouze dvě možnosti: (Dobře si zde všimněte, že přechodem na krok  $t+1$  podle předchozího bodu skutečně dochází k jediné změně situace, kdy jsou teď navíc povoleny cesty přes vnitřní vrchol  $t$ .)
  - Buď je cesta délky  $d[i][j]$  z předchozího kroku stále nejlepší (tj. nově povolený vrchol  $t$  nám nepomůže),
  - nebo cestu vylepšíme spojením přes nově povolený vrchol  $t$ , čímž získáme menší vzdálenost  $d[i][t] + d[t][j]$ . (Nakreslete si obrázek!)

**Poznámka:** V praktické implementaci pro symbol  $\infty$  použijeme velkou konstantu, třeba  $\text{MAX\_INT}/2$ . (Nelze použít přímo  $\text{MAX\_INT}$ , neboť by pak došlo k aritmetickému přetečení.)

### Algoritmus 8.6. *Výpočet metriky grafu*

Tento algoritmus, pro graf s  $N$  vrcholy daný maticí sousednosti  $G$ , vypočte celou jeho metriku  $d$ .

```
vstup: matice sousednosti  $G[] []$  grafu na  $N$  vrcholech,  
kde  $G[i][j]=1$  pro hranu mezi  $i, j$  a  $G[i][j]=0$  jinak;  
for (i=0; i<N; i++) for (j=0; j<N; j++)  
    d[i][j] = (i==j?0: (G[i][j]? 1: MAX_INT/2));  
for (t=0; t<N; t++) {  
    for (i=0; i<N; i++) for (j=0; j<N; j++)  
        d[i][j] = min(d[i][j], d[i][t]+d[t][j]);  
}
```

Poznámka: Algoritmus 8.6 je implementačně velmi jednoduchý (vždyť se celý jeho kód vešel na 5 přehledných řádků) a provede zhruba  $N^3$  kroků pro výpočet celé metriky. Jeho jedinou (ale velkou) nevýhodou je, že vzdálenosti mezi všemi dvojicemi vrcholů je třeba počítat najednou. V praktických situacích však obvykle požadujeme zjištění vzdálenosti mezi jednou dvojicí vrcholů, a pak celý zbytek výpočtu je k ničemu...

### Úlohy k řešení

(8.2.1) Vypočtěte dle Algoritmu 8.6 metriku kružnice  $C_4$ .

## 8.3 Vážená (ohodnocená) vzdálenost

V dalším textu se již budeme věnovat cestám v grafech s obecně “dlouhými” hranami.

**Definice 8.7.** *Vážený graf* je graf  $G$

spolu s ohodnocením  $w$  hran reálnými čísly  $w : E(G) \rightarrow \mathbb{R}$ .

*Kladně vážený graf*  $G, w$  je takový, že  $w(e) > 0$  pro všechny hrany  $e$ .

**Definice:** Mějme (kladně) vážený graf  $G, w$ . Délkou váženého sledu  $S = v_0, e_1, v_1, e_2, v_2, \dots, e_n, v_n$  v  $G$  myslíme součet

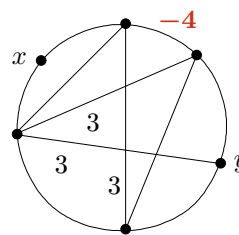
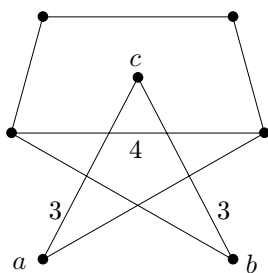
$$d_G^w(S) = w(e_1) + w(e_2) + \dots + w(e_n).$$

*Váženou vzdáleností* v  $G, w$  mezi dvěma vrcholy  $u, v$  pak myslíme

$$d_G^w(u, v) = \min\{d_G^w(S) : S \text{ je sled s konci } u, v\}.$$

**Lema 8.8.** *Vážená vzdálenost v kladně vážených grafech také splňuje trojúhelníkovou nerovnost.*

**Komentář:** Podívejme se na následující graf vlevo. (Čísla u hran udávají jejich váhy, hrany bez čísel mají váhu 1.) Vážená vzdálenost mezi vrcholy  $a, c$  je 3, mezi  $b, c$  také 3. Jaká je vzdálenost mezi vrcholy  $a, b$ ? Ne, není to 6, ale najdeme kratší cestu délky 5.



V druhém příkladě vpravo je uvedena i hrana se zápornou vahou  $-4$ . Nejkratší cesta mezi vrcholy  $x, y$  tak má délku  $-2$ , ale pokud vezmeme sled, který hranu váhy  $-4$  vícekrát zopakuje, dostaneme se na libovolně nízkou zápornou délku. To je samozřejmě nesmyslné, a proto se takovému problému radši vyhýbáme **zákazem záporných vah hran**.

### Úlohy k řešení

(8.3.1) Jaká je nejdelší vzdálenost mezi dvěma vrcholy v předchozím obrázku vlevo?

(8.3.2) O kterém vrcholu v předchozím obrázku vlevo se dá říci, že má “centrální pozici”, tj. že je z něj do všech ostatních vrcholů nejbližší? Jaká je z něj největší vzdálenost do ostatních vrcholů?

## 8.4 Hledání nejkratší cesty

Pro nalezení nejkratší (vážené) cesty mezi dvěma vrcholy kladně váženého grafu se používá známý tzv. *Dijkstrův algoritmus*.

**Poznámka:** Dijkstrův algoritmus je sice složitější než Algoritmus 8.6, ale na druhou stranu je výrazně rychlejší, pokud nás zajímá jen nejkratší vzdálenost z jednoho vrcholu místo všech dvojic vrcholů.

Zrovna tento algoritmus se například používá při vyhledávání vlakových či autobusových spojení. Pravděpodobně se i vy někdy dostanete do situace, kdy budete nejkratší cestu hledat, proto si tento algoritmus zapamatujte.

### Dijkstrův algoritmus

- Je variantou procházení grafu (skoro jako do šířky), kdy pro každý nalezený vrchol ještě máme *proměnnou udávající vzdálenost* – délku nejkratšího sledu (od počátku), kterým jsme se do tohoto vrcholu zatím dostali.
- Z úschovny nalezených vrcholů vždy vybíráme **vrchol s nejmenší vzdáleností** (mezi uschovanými vrcholy) – do takového vrcholu se už lépe dostat nemůžeme, protože všechny jiné cesty by byly dle výběru delší.
- Na konci zpracování tyto proměnné vzdálenosti udávají správně nejkratší vzdálenosti z počátečního vrcholu do ostatních.

### Algoritmus 8.9. *Dijkstrův pro nejkratší cestu v grafu*

*Tento algoritmus nalezne nejkratší cestu mezi vrcholy u a v kladně váženého grafu G, daného seznamem sousedů vrcholů.*

```
vstup: graf na N vrcholech daný seznamem sousedů sous [] [] a wh [] [],
      kde sous[i][0], ..., sous[i][st[i]-1] jsou sousedé vrcholu i stupně st[i]
      a hrana z i do sous[i][k] má délku wh[i][k]>0;
vstup: u, v, kde hledáme cestu z u do v;

// stav[i] udává zpracovanost vrcholu, vzdal[i] zatím nalezenou vzdálenost
for (i=0; i<=N; i++) { vzdal[i] = MAX_INT; stav[i] = 0; }
vzdal[u] = 0;
while (stav[v]==0) {
  for (i=0, j=N; i<N; i++)
    if (stav[i]==0 && vzdal[i]<vzdal[j]) j = i;
  // zde jsme našli nejbližší nezpracovaný vrchol j, ten teď zpracujeme
  if (vzdal[j]==MAX_INT) return NENI_CESTA;
  stav[j] = 1;
  for (k=0; k<st[j]; k++)
    if (vzdal[j]+wh[j][k]<vzdal[sous[j][k]]) {
      pri[sous[j][k]] = j;
      vzdal[sous[j][k]] = vzdal[j]+wh[j][k];
    }
  // pole pri[...] uchovává, odkud jsme se do kterého vrcholu dostali
}
výstup 'Cesta délky vzdal[v], uložená pozpátku v poli pri []';
```

**Poznámka:** Uvědomme si, že pokud necháme tento algoritmus proběhnout až do zpracování všech vrcholů, získáme v vzdal[i] nejkratší vzdálenosti z počátečního vrcholu do všech ostatních vrcholů. Všimněme si dále, že Algoritmus 8.9 počítá stejně dobře nejkratší cestu i v **orientovaném grafu**.

Důkaz správnosti Algoritmu 8.9.

Klíčovým faktem k důkazu správnosti algoritmu je to, že v každém kroku (počínaje stavem po prvním průchodu cyklem `while()`) proměnná `vzdal[i]` udává nejkratší vzdálenost z vrcholu `u` do vrcholu `i` při cestě pouze po vnitřních vrcholech `x`, jejichž `stav[x]==1` (po zpracovaných vrcholech):

- V prvním kroku algoritmu je jako vrchol ke zpracování vybrán první `j=u` a potom jsou jeho sousedům upraveny vzdálenosti od `u` podle délek hran `z u`.
- V každém dalším kroku je vybrán jako vrchol `j` ke zpracování ten, který má ze všech nezpracovaných vrcholů nejkratší nalezenou vzdálenost od počátku `u`. To ale znamená, že žádná kratší cesta už do `j` nevede, neboť každá oklika přes jiné nezpracované vrcholy musí být delší dle výběru `j`. (V tomto bodě potřebujeme **nezápornost ohodnocení** `wh[] []`.) Naopak můžeme podle délek hran vycházejících z `j` “vylepšit” cesty do jeho sousedů.

Závěrem zbývá ověřit, že nalezená nejkratší cesta z `u` do `v` je pozpátku uložená v poli `pri[]`, tj. předposlední vrchol před `v` je `pri[v]`, předtím `pri[pri[v]]`, atd...  $\square$

**Fakt:** Celkový počet kroků potřebný v Algoritmu 8.9 k nalezení nejkratší cesty z `u` do `v` je zhruba  $N^2$ , kde  $N$  je počet vrcholů grafu. Na druhou stranu, při lepší implementaci úschovny nezpracovaných vrcholů (třeba haldou s nalezenou vzdáleností jako klíčem) lze dosáhnout i mnohem rychlejšího běhu tohoto algoritmu na řídkých grafech – času zhruba úměrného počtu hran grafu.

**Komentář:** Vysvětleme si nyní podrobně srovnání obou našich algoritmů. Algoritmus 8.6 pro výpočet metriky je implementačně jednodušší a přímočařejší a hodí se tam, kde potřebujeme spočítat všechny dvojice vzdáleností v grafu najednou. Velkou nevýhodou však je, že vždy musíme provést  $N^3$  kroků celého výpočtu, i když počítáme vzdálenost jen dvou blízkých vrcholů.

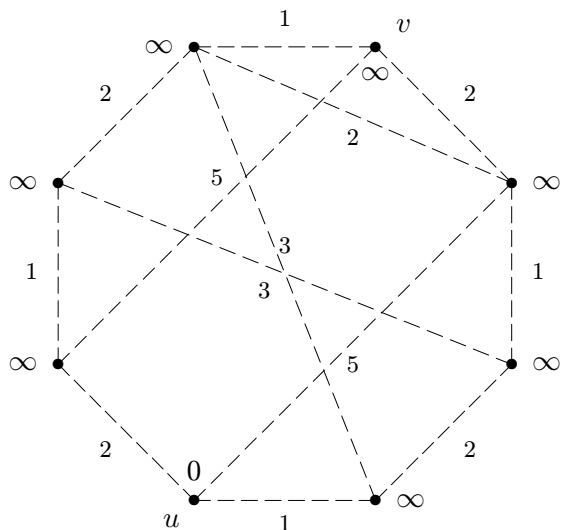
Dijkstrův algoritmus na druhou stranu počítá mnohem rychleji, pokud nás zajímá jen vzdálenost z jednoho vrcholu (zhruba  $N^2$  nebo i méně kroků), a dokonce běží ještě rychleji, pokud nás zajímá jen vzdálenost dvou blízkých vrcholů – tehdy totiž můžeme prohledávání ukončit i jen na malé části celého grafu. To jsou přímo ideální vlastnosti, které bychom od takového algoritmu očekávali.

### Rozšiřující studium

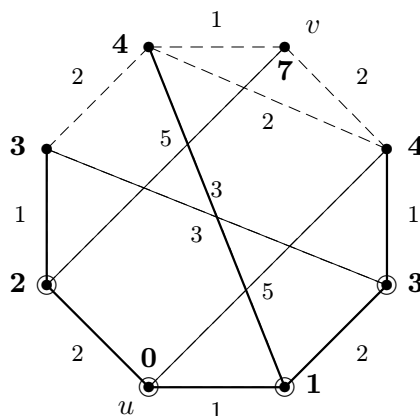
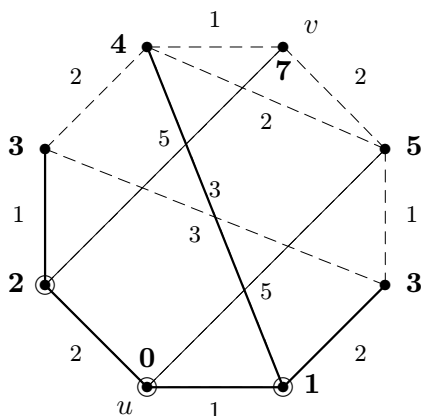
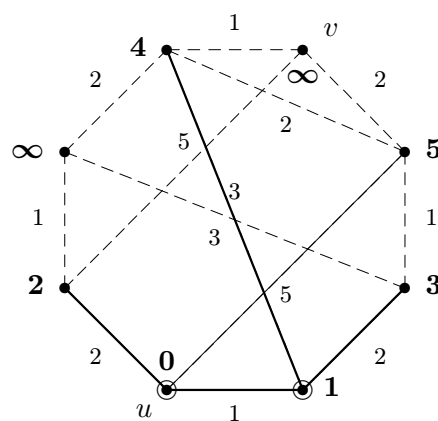
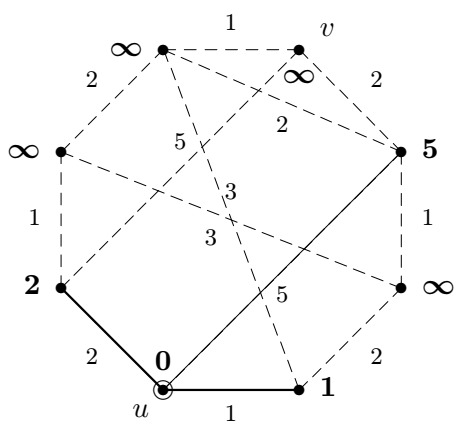
*Vzdálenostem v grafech se více věnují [5, Oddíly 3.2,3]. Dijkstrův algoritmus je velmi oblíbeným tématem mnoha učebnic programování, a proto jej není třeba nijak zvlášť hledat.*

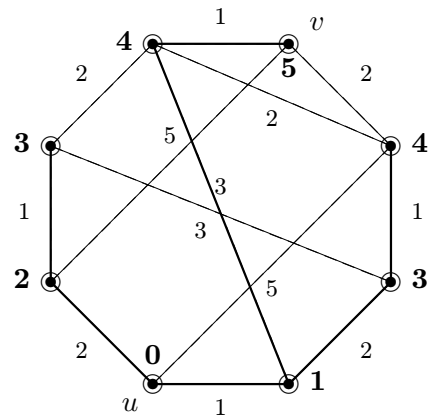
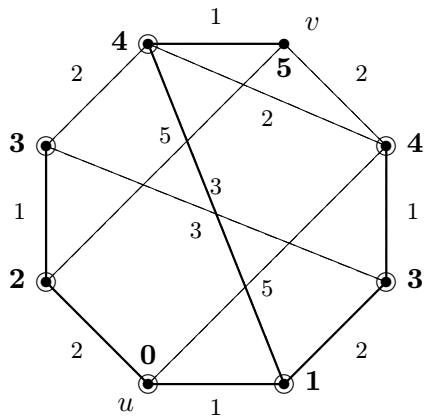
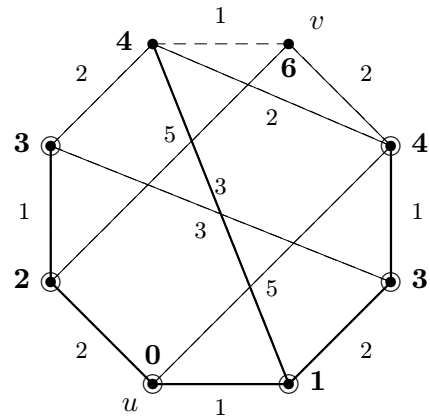
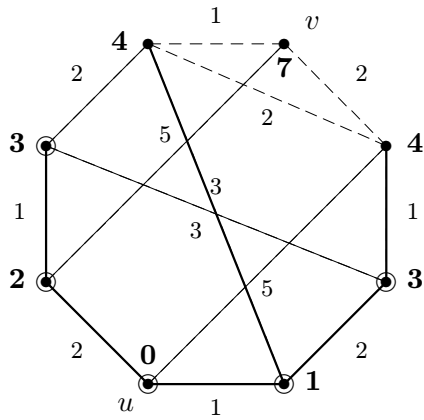
## 8.5 Cvičení: Příklady o grafových vzdálenostech

**Příklad 8.10.** Ukázka běhu Dijkstrova Algoritmu 8.9 pro nalezení nejkratší cesty mezi vrcholy `u, v` v následujícím grafu.



U tohoto algoritmu se vlastně jedná a specifickou variantu procházení grafu. Proto použijeme stejné obrázkové značení jako v Příkladě 7.8 a navíc pro každý vrchol zakreslíme jeho okamžitou dočasnou vzdálenost  $vzda1[x]$ . (Tj. zpracované vrcholy budeme značit kroužkem a hrany plnou čarou.) Jednotlivé kroky následují:

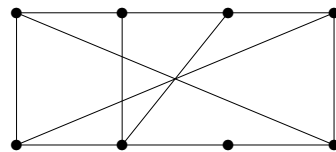
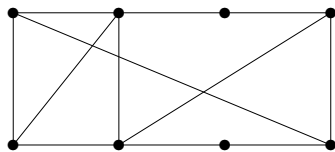




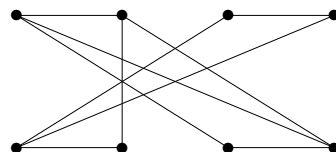
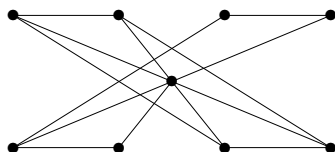
Z průběhu algoritmu vidíme, že již ve třetím kroku jsme určili dočasnou vzdálenost z  $U$  do  $v$  na 7, ale to nebyla ta nejkratší. Nakonec po proběhnutí všech kroků algoritmu vzdálenost  $u, v$  poklesla až na optimálních 5. Nejkratší cesta je naznačená tlustými čarami. Pamatujte proto, že Dijkstrův algoritmus musíte provádět vždy tak dlouho, dokud se konečně nezpracuje cílový vrchol.  $\square$

### Úlohy k řešení

(8.5.1) Jaká je největší vzdálenost mezi dvěma vrcholy v každém z následujících dvou grafů? Vyznačte tyto dva nejvzdálenější vrcholy.

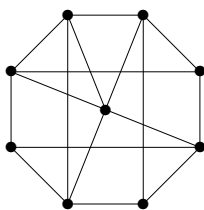


(8.5.2) Jaká je největší vzdálenost mezi dvěma vrcholy v každém z následujících dvou grafů? Vyznačte tyto dva nejvzdálenější vrcholy.

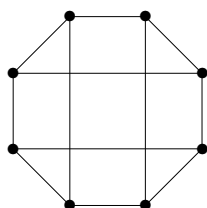




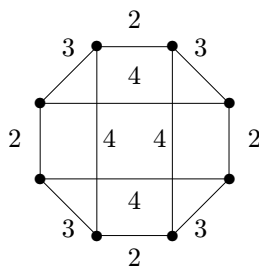
(8.5.3) Kolik (neuspořádaných) dvojic vrcholů v následujícím grafu má vzdálenost právě 3?



(8.5.4) Kolik nejméně hran musíme přidat do následujícího grafu, aby největší vzdálenost mezi dvěma vrcholy byla 2? Zdůvodněte.



(8.5.5) Jaká je největší vzdálenost mezi dvojicí vrcholů v tomto váženém grafu?



(8.5.6) Kolik nejvíce vrcholů může mít graf, který má všechny vrcholy stupně 3 a největší vzdálenost mezi dvěma vrcholy je 2?

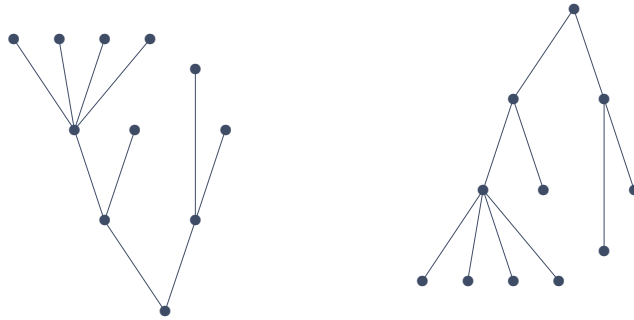
\* (8.5.7) Jaká největší vzdálenost může být mezi dvěma vrcholy kružnice délky 9, jejíž hrany jsou ohodnoceny vzdálenostmi 1, 2, ..., 8, 9 v nějakém (libovolném) pořadí?

\* (8.5.8) Představme si graf, jehož vrcholy jsou všechna přirozená čísla od 2 do 15 a hrany spojují právě ty dvojice vrcholů, které jsou soudělné jako čísla. Přitom délka hrany je vždy rovna největšímu společnému děliteli. (Například 4, 5 nejsou spojené a 8, 12 jsou spojené hranou délky 4.) Pomineme-li izolované vrcholy 11 a 13, jaká je největší vzdálenost mezi zbylými vrcholy tohoto grafu?

## 9 Stromy a les

### Úvod

Jedním ze základních, a patrně nejjednodušším, typem grafů jsou takzvané stromy. Jedná se o souvislé grafy bez kružnic. Přes svou (zdánlivou) jednoduchost mají stromy bohatou strukturu a především množství vlastních aplikací. Patrně nejstarší motivací pojmu stromu jsou rodokmeny, jejichž původ sahá daleko před vznik teorie grafů. Proto také mnoho pojmů týkajících se stromů je motivováno rodokmeny. Co vidíme na následujících dvou obrázcích?



O důležitosti souvislosti grafu jsme pojednali dříve. Se stromy jako se souvislými (pod)grafy bez kružnic se pak setkáváme nejvíce v situacích, kde absence kružnic vyplývá z podstaty věci, nebo kde jsou kružnice zcela nežádoucí. To jsou různá schémata, datové struktury či již zmíněné rodokmeny. Stromy například přirozeně popisují různé hierarchické struktury. Mimo popisy struktur je ještě jedna důležitá oblast aplikací stromů, týkající se tzv. koster a problému hledání minimální kostry v grafu.

### Cíle

Úkolem této lekce je hlavně definovat pojem stromu a ukázat základní vlastnosti stromů včetně jejich zdůvodnění. Podrobněji jsou probrány kořenové a uspořádané stromy a práce s nimi. Dále je zaveden pojem kostry grafu a důležitý problém minimální kostry, včetně jednoduchého “hladového” algoritmu pro jeho řešení.

## 9.1 Základní vlastnosti stromů

Začneme definicemi lesa a stromů. Jelikož i smyčky a násobné hrany v multigrafech jsou považovány za kružnice délek 1 a 2, v lesích a stromech se nenacházejí. Bavíme se zde tudíž pouze o jednoduchých grafech.

**Definice:** *Les* je jednoduchý graf bez kružnic.

### Definice 9.1. *Strom*

je jednoduchý souvislý graf  $T$  bez kružnic.

**Fakt.** Komponenty souvislosti lesa jsou stromy. Jeden vrchol (bez hran) je také strom.

Dále si ukážeme přehled základních vlastností stromů, včetně jejich zdůvodnění, která jsou poměrně jednoduchá. Následující vlastnosti dokonce plně popisují stromy a mohou tudíž být použity místo uvedené Definice 9.1. Jedná se sice o poněkud obtížný teoretický oddíl, ale alespoň zběžné porozumění uvedeným vlastnostem je potřebné pro další práci se stromy a s jinými strukturami založenými na stromech.

**Lema 9.2.** *Strom s více než jedním vrcholem má nějaký vrchol stupně 1.*

**Důkaz.** Souvislý graf s více než jedním vrcholem nemůže mít vrchol stupně 0. Proto vezmeme strom  $T$  a v něm libovolný vrchol  $v$ . Sestrojíme nyní co nejdelší sled  $S$  v  $T$  začínající ve  $v$ :  $S$  začne libovolnou hranou vycházející z  $v$ . V každém dalším vrchole  $u$ , do kterého se dostaneme a má stupeň větší než 1, lze pak pokračovat sled  $S$  další novou hranou. Uvědomme si, že pokud by se ve sledu  $S$  poprvé zopakoval některý vrchol, získali bychom kružnici, což ve stromu nelze. Proto sled  $S$  musí jednou skončit v nějakém vrchole stupně 1 v  $T$ .  $\square$

**Věta 9.3.** *Strom na  $n$  vrcholech má právě  $n - 1$  hran pro  $n \geq 1$ .*

**Důkaz.** Toto tvrzení dokážeme indukcí podle  $n$ . Strom s jedním vrcholem má  $n - 1 = 0$  hran. Nechť  $T$  je strom na  $n > 1$  vrcholech. Podle Lematu 9.2 má  $T$  vrchol  $v$  stupně 1. Označme  $T' = T - v$  graf vzniklý z  $T$  odebráním vrcholu  $v$ . Pak  $T'$  je také souvislý bez kružnic, tudíž strom na  $n - 1$  vrcholech. Dle indukčního předpokladu  $T'$  má  $n - 1 - 1$  hran, a proto  $T$  má  $n - 1 - 1 + 1 = n - 1$  hran.  $\square$

**Věta 9.4.** *Mezi každými dvěma vrcholy stromu vede právě jediná cesta.*

**Důkaz.** Jelikož strom  $T$  je souvislý dle definice, mezi libovolnými dvěma vrcholy  $u, v$  vede nějaká cesta. Pokud by existovaly dvě cesty mezi  $u, v$ , tak jejich sjednocení by bylo uzavřeným tahem v  $T$  a po “zkrácení” všech opakovaných vrcholů v tomto tahu by vznikla kružnice v  $T$ , což je spor s předpokladem. Proto cesta mezi  $u$  a  $v$  existuje jen jedna.  $\square$

**Důsledek 9.5.** *Přidáním jedné nové hrany do stromu vznikne právě jedna kružnice.*

**Důkaz.** Nechť mezi vrcholy  $u, v$  ve stromu  $T$  není hrana. Přidáním hrany  $e = uv$  vznikne právě jedna kružnice z  $e$  a jediné cesty mezi  $u, v$  v  $T$  podle Věty 9.4.  $\square$

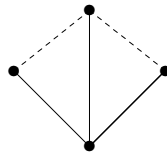
**Věta 9.6.** *Strom je minimální souvislý graf (na daných vrcholech).*

**Důkaz.** Strom je souvislý podle definice. Pokud by souvislý graf měl kružnici, zůstal by souvislý i po vypuštění některé hrany té kružnice. Proto minimální souvislý graf je stromem. Naopak, pokud by vypuštěním hrany  $e = uv$  ze stromu  $T$  vznikl souvislý graf, pak by mezi  $u, v$  v  $T$  existovaly dvě cesty (dohromady kružnice) – hrana  $e$  a jiná cesta v  $T - e$ . To je ve sporu s Větou 9.4. Proto je strom minimálním souvislým grafem na daných vrcholech.  $\square$

Závěrem si pro správné pochopení základních vlastností stromů vyřešíme následující (nepříliš jednoduchý) příklad.

**Příklad 9.7.** *Kolik nejvýše kružnic vznikne v grafu, který vytvoříme ze stromu přidáním dvou hran?*

Přidáním jedné hrany do stromu  $T$  vznikne jedna kružnice dle Důsledku 9.5. Druhá hrana vytvoří nejméně ještě jednu kružnici ze stejných důvodů, ale může vytvořit i dvě další kružnice, jako třeba v následujícím grafu, kde strom  $T$  je vyznačen plnými čarami a dvě přidané hrany čárkovaně.



Každá z přidaných dvou hran vytvoří vlastní trojúhelník a navíc ještě vznikne kružnice délky 4 procházející oběma z přidaných hran.

Na druhou stranu chceme ukázat, že více než 3 kružnice vzniknout nemohou po přidání dvou hran  $e, f$  do stromu  $T$ : Podle Důsledku 9.5 vznikne jen jedna kružnice procházející hranou  $e$  a neobsahující  $f$ , stejně tak jedna kružnice procházející  $f$  a neobsahující  $e$ . Nakonec stačí nahlédnout, že je nejvýše jedna možná kružnice procházející oběma hranami  $e, f$ , neboť jinak by už ve stromě  $T$  musela být kružnice.  $\square$

## Úlohy k řešení

(9.1.1) *Kolik je neisomorfních lesů na třech vrcholech? Nakreslete si je.*

(9.1.2) *Kolik je neisomorfních stromů na čtyřech vrcholech? Nakreslete si je.*

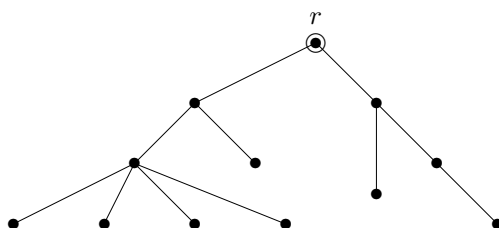
\*(9.1.3) *Najdete graf s dvěma kružnicemi, z něhož lze odebráním jedné hrany vytvořit strom? Zdůvodněte a případně nakreslete.*

## 9.2 Kořenové stromy

Při mnoha použitích stromů se ke stromu jako grafu samotnému ještě váží dodatečné informace, jako třeba vyznačený jeden vrchol, tzv. kořen stromu, ze kterého celý strom “vyrůstá”. Typickým příkladem jsou různé (acyklické) datové struktury, ve kterých je vyznačený vrchol – kořen, referován jako “začátek” uložených dat. Jinak třeba evoluční stromy druhů v biologii mají za kořen jejich společného (dávného) předchůdce. Kořenové stromy mají také tradiční motivaci v rodokmenech a z toho vychází jejich běžná terminologie.

**Definice 9.8.** *Kořenovým stromem* je strom  $T$  spolu s vyznačeným *kořenem*  $r \in V(T)$ , zkráceně zapsaný dvojicí  $T, r$ .

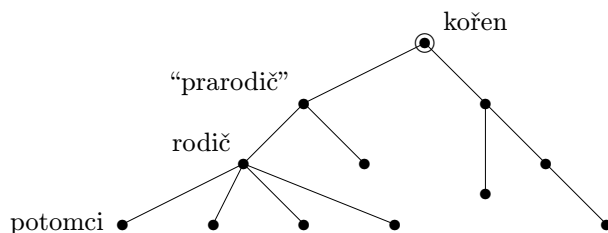
**Komentář:** Příklad kořenového stromu je na následujícím obrázku:



Zajímavostí je, že v informatice stromy většinou rostou od kořene směrem dolů. (Však také nejsme v biologii. . .)

**Definice:** Mějme kořenový strom  $T, r$  a v něm vrchol  $v$ . Označme  $u$  souseda  $v$  na cestě směrem ke kořeni  $r$ . Pak je  $u$  nazýván *rodičem*  $v$  a  $v$  je nazýván *potomkem*  $u$ .

**Komentář:** Kořen nemá žádného rodiče. V přirozeně přeneseném významu se u kořenových stromů používají pojmy prarodič, předchůdce, následovník, sourozenci, atd. Zběžná ilustrace použití těchto pojmů je na následujícím schématu stromu.



Často se také setkáte v kořenových stromech s označováním “otec–syn” místo rodič–potomek. My jsme takové označení nepoužili proto, že by (hlavně v zemích na západ od nás) mohlo být považováno za “sexistické”.

**Definice:** Vrchol stupně 1 v libovolném stromu nazýváme *listem*.

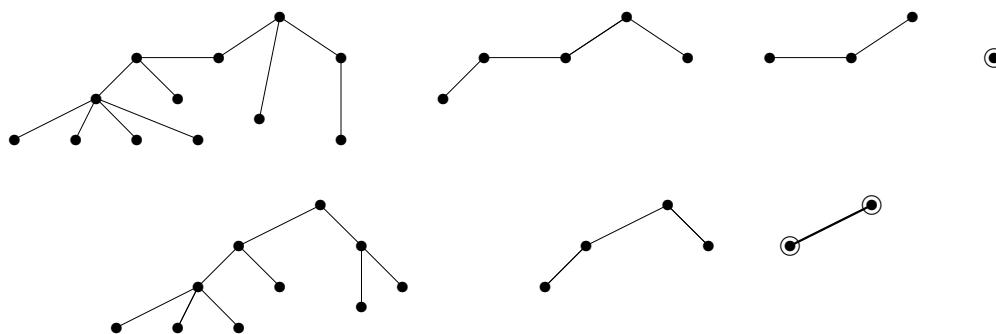
**Komentář:** Pozor, i kořen stromu může být listem, pokud má stupeň 1, ale obvykle se to tak neříká. List kořenového stromu, který není kořenem, nemá potomky.

Občas se můžeme dostat do situace, kdy k danému stromu potřebujeme zvolit kořen, aby jeho volba byla jednoznačná, tj. aby bylo zaručeno, že pro dva isomorfní stromy nezávisle zvolíme tentýž kořen. K tomu nám dopomůže následující názorná definice centra.

**Definice:** *Centrem stromu*  $T$  rozumíme buď vrchol nebo hranu nalezenou v  $T$  následujícím postupem:

- Pokud má strom  $T$  jeden vrchol, je to jeho centrum. Pokud má strom  $T$  dva vrcholy, je jeho centrem hrana spojující tyto dva vrcholy.
- Jinak vytvoříme menší strom  $T' \subseteq T$  vypuštěním všech listů  $T$  najednou. Je zřejmé, že  $T'$  je neprázdný, a vracíme se na předchozí bod. Získané (rekurzivně) centrum  $T'$  je zároveň centrem  $T$ .

**Příklad 9.9.** Ilustrací definice centra jsou následující dva postupy nalezení centra (zleva doprava):



V prvním stromě získáme kořen jako jediný vrchol po třech rekurzivních krocích odebrání listů. V druhém stromě je kořenem hrana, kterou získáme po dvou rekurzivních krocích.  $\square$

**Fakt.** Pokud chceme danému (abstraktnímu) stromu přiřadit jednoznačně kořen, je nejlepší jej přiřadit centru stromu. Speciálně, pokud je centrem hrana, bude kořenem nový vrchol “rozdělující” tuto hranu na dvě. Viz předchozí příklad:



Další dodatečnou informací často vázanou ke kořenovým stromům je nějaké uspořádání potomků každého vrcholu, jako třeba seřazení potomků v rodokmenech podle jejich data narození. To formalizujeme následující definicí.

**Definice:** Kořenový strom  $T, r$  je *uspořádaný*, pokud je pro každý jeho vrchol jednoznačně dáno pořadí jeho potomků (“zleva doprava”).

Uspořádaný kořenový strom se také nazývá *pěstovaný strom*.

**Komentář:** Uspořádaný kořenový strom si jinak také můžeme představit jako strom s vyznačeným kořenem a pevně zvoleným nakreslením v rovině bez křížení hran. Nakreslení hran potomků vzhledem k hraně rodiče pak udává (ve zvolené orientaci) pořadí potomků. Tento pohled vede k názvu pěstovaný strom.

Uspořádání potomků vrcholu ve stromu je přirozeně požadováno v mnoha praktických situacích. Například ve stromových datových strukturách jsou často potomci explicitně seřazení podle daného klíče, jako třeba ve vyhledávacích binárních stromech. I v případech, kdy uspořádání potomků ve stromě není dáno, je možné jej jednoznačně definovat, jak uvidíme v následující části.

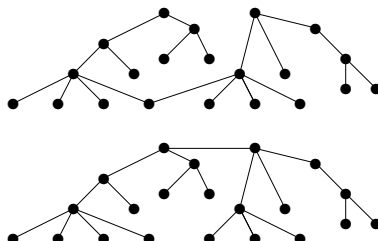
## Úlohy k řešení

**(9.2.1)** Binární vyhledávací strom je uspořádaný kořenový strom, který se pro daná data obvykle tvoří následovně: První prvek dat se uloží do kořene. Každý další příchozí prvek,

pokud má menší klíč než kořen, uloží se rekurzivně do levého podstromu, pokud větší, tak do pravého podstromu. Vytvořte binární vyhledávací strom pro danou posloupnost dat s klíči

11, 15, 9, 5, 13, 10, 20, 21, 1, 6.

(9.2.2) Určete centra následujících dvou stromů:



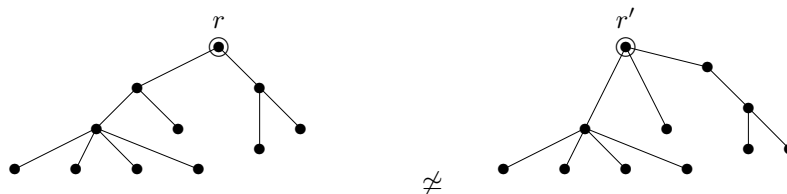
(9.2.3) Mějme libovolný strom s 33 vrcholy. Kolik jeho listů postupně odebereme, než určíme centrum? (Je to jednoznačné?)

### 9.3 Isomorfismus stromů

Jelikož stromy jsou speciálním případem grafů, je isomorfismus stromů totéž co isomorfismus grafů obecně. Avšak na rozdíl od grafů, kdy je určení isomorfismu algoritmicky (nakonec i ručně) těžký problém, pro isomorfismus stromů existuje efektivní postup, který si ukážeme. Nejprve si uvedeme restriktivnější (tj. vyžadující více shody) verze definice isomorfismu pro kořenové a uspořádané stromy.

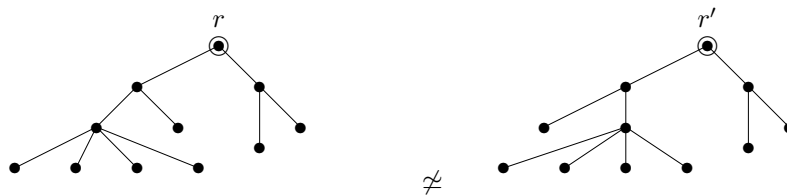
**Definice:** (Dva stromy jsou isomorfní pokud jsou isomorfní jako grafy.) Dva kořenové stromy  $T, r$  a  $T', r'$  jsou *isomorfní* pokud existuje isomorfismus mezi stromy  $T$  a  $T'$ , který kořen  $r$  zobrazuje na kořen  $r'$ .

**Komentář:** Například následující dva (isomorfní) stromy *nejsou isomorfní* coby kořenové stromy.



**Definice:** Dva uspořádané kořenové (pěstované) stromy jsou isomorfní pokud je mezi nimi isomorfismus kořenových stromů, který navíc zachovává pořadí potomků všech vrcholů.

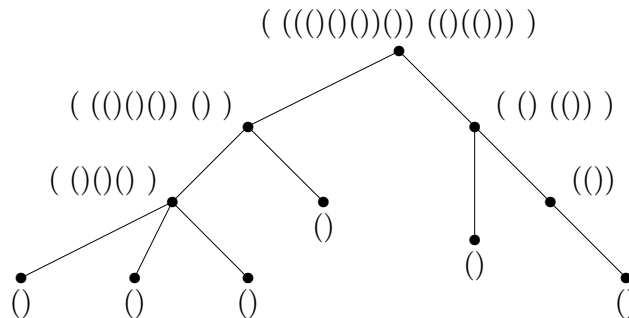
**Komentář:** Například následující dva (isomorfní) kořenové stromy *nejsou isomorfní* coby uspořádané kořenové stromy, neboť pořadí potomků levého syna kořene se liší.



## Kódování uspořádaných kořenových stromů

Uspořádanému kořenovému stromu lze snadným postupem přiřadit řetězec vnořených závorek, který jej plně popisuje. (Možná jste se již s touto jednoduchou korespondencí závorek a kořenových stromů setkali, třeba při sémantické analýze matematických výrazů.)

Definice:



*Kód* uspořádaného kořenového stromu se spočítá rekurzivně z kódů všech podstromů kořene, seřazených v daném pořadí a uzavřených do páru závorek.

Poznámka: Místo znaků '(' a ')' lze použít i jiné symboly, třeba '0' a '1'.

Klíčovým a snadno zdůvodnitelným faktem o kódech pěstovaných stromů je toto tvrzení:

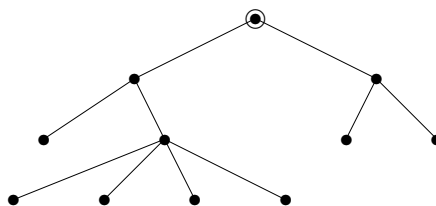
**Lema 9.10.** *Dva uspořádané kořenové (pěstované) stromy jsou isomorfní právě když jejich kódy získané podle předchozího popisu jsou shodné řetězce.*

**Fakt.** Je-li dán kód  $s$  uspořádaného kořenového stromu, příslušný strom nakreslíme následujícím postupem:

- Při přečtení znaku '(' na začátku spustíme pero na papír, do kořene.
- Při každém dalším přečtení znaku '(' nakreslíme hranu do následujícího potomka současného vrcholu.
- Při každém přečtení znaku ')' se perem vrátíme do rodiče současného vrcholu, případně zvedneme pero, pokud už jsme v kořeni.

**Příklad 9.11.** *Nakreslete jako pěstovaný strom ten odpovídající závorkovému kódu*

$((0(0(0(0(0))))(0(0))).$



□

Při určování isomorfismu obecných stromů použijeme Lema 9.10 pro jejich jednoznačnou reprezentaci uspořádanými kořenovými stromy, ve které kořen volíme v centru

a potomky seřadíme podle jejich kódů vzestupně lexikograficky (tj. abecedně). Jinak se dá také říci, že kód přiřazený stromu  $T$  je lexikograficky nejmenší ze všech kódů uspořádaných stromů vzniklých z  $T$  s kořenem v jeho centru. Takový kód je zřejmě určující vlastností stromu  $T$  jako takového, a proto správnost následujícího algoritmu můžeme snadno nahlédnout.

**Algoritmus 9.12. *Určení isomorfismu dvou stromů.***

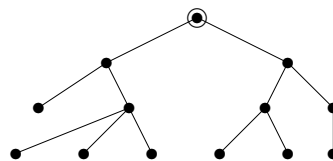
*Pro dané dva obecné stromy  $T$  a  $U$  implementujeme algoritmus zjišťující isomorfismus  $T \simeq? U$  následovně v symbolickém zápise:*

```
Vstup < stromy  $T$  a  $U$ ;
for ( $X=T,U$ ) { // určení center daných stromů pro kořeny
   $x = \text{centrum}(X)$ ;
  if ( $x$  je jeden vrchol)  $r = x$ ;
  else nový  $r$ , nahraď hranu  $x=uv$  hranami  $ru, rv$ ;
   $k[X] = \text{minimalni\_kod}(X,r)$ ;
}
if ( $k[T]==k[U]$  jako řetězce) printf("Jsou isomorfní.");
else printf("Nejsou isomorfní.");
exit;

Funkce minimalni_kod(strom  $X$ , vrchol  $r$ ) {
  if ( $X$  má jeden vrchol) return "()";
   $Y[1\dots d] = \{\text{souvislé komponenty } X-r, \text{ tj. podstromy kořene } r\}$ ;
   $s[1\dots d] = \{\text{kořeny podstromů } Y[i] \text{ v odpovídajícím pořadí}\}$ ;
  for ( $i=1, \dots, d$ )
     $k[i] = \text{minimalni\_kod}(Y[i],s[i])$ ;
  sort lexikograficky (abecedně) podle klíče  $k[1] \leq k[2] \leq \dots \leq k[d]$ ;
  return (" $+k[1]+\dots+k[d]+$ ")";
}
```

**Úlohy k řešení**

(9.3.1) Odvoďte správný závorkový kód pro tento pěstovaný strom:



(9.3.2) Nakreslete pěstovaný strom odpovídající závorkovému kódu

$((((())())((())()))).$

**9.4 Kostry grafů**

Kromě stromů samotných se zabýváme i stromy, které jsou obsaženy jako podgrafy ve větších grafech.

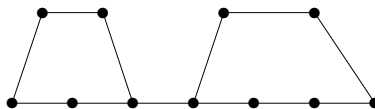
**Definice 9.13. *Kostrou*** souvislého grafu  $G$  je podgraf v  $G$ , který je sám stromem a obsahuje všechny vrcholy grafu  $G$ .



**Komentář:** Kostrou stromu je strom sám. Na druhou stranu kostrou kružnice  $C_n$  je každá z  $n$  cest vzniklých z  $C_n$  vypuštěním jedné hrany. Složitější grafy mají pak ještě více možných koster.

**Poznámka:** Pojem kostry lze definovat i pro nesouvislé grafy, pak se kostrou myslí les, jehož stromy jsou kostrami jednotlivých komponent.

**Příklad 9.14.** *Kolik různých koster má tento graf?*



Podívejme se na kostru grafu takto – jaké hrany z grafy vymažeme, aby zbyl strom? Zajisté musíme vymazat některou hranu z první kružnice (5 možností) a některou hranu z druhé kružnice (6 možností). Na druhou stranu to v tomto jednoduchém příkladě už stačí, vždy pak zůstane strom. Výběr vymazané hrany z první kružnice je nezávislý na druhé kružnici (jsou disjunktní), a proto dle principu nezávislých výběrů máme  $5 \cdot 6 = 30$  možností vybrat dvě hrany k vymazání. Celkem tedy vyjde 30 koster.  $\square$

Počet všech koster grafu lze obecně spočítat determinanem jisté matice, ale to už přesahuje rámec našeho textu. Zde si jen uvedeme následující fakt, jehož důkaz je také nad rámec textu.

**Fakt.** Úplný graf  $K_n$  má přesně  $n^{n-2}$  koster.

Kromě teoretických “hrátek” mají kostry grafů jedno důležité praktické použití, jež ukážeme nyní.

**Komentář:** Dříve jsme uvažovali spojení v grafech cestami jdoucími z jednoho místa do druhého, ale nyní se podíváme na jiný způsob “propojování” všech vrcholů grafu najednou: Naším cílem je najít minimální souvislý podgraf daného grafu, tedy minimální způsob propojení (v daných podmínkách), ve kterém existují cesty mezi každou dvojicí vrcholů. Podle Věty 9.6 je tímto minimálním propojením strom – kostra našeho grafu.

Tak je třeba uvažováno propojení domů elektrickým rozvodem, propojení škol internetem, atd. Zde nás ani tak nezajímají délky cest mezi propojenými body, ale hlavně celková délka či cena vedení/spojení, které musíme postavit. Vstupní graf nám udává všechny možné realizovatelné propojky s jejich cenami. Takto postavené zadání pak vede na následující teoretický problém.

### **Problém 9.15. Minimální kostry (MST)**

*Je dán souvislý vážený graf  $G, w$  s nezáporným ohodnocením hran  $w$ . Otázkou je najít kostru  $T$  v  $G$ , která má nejmenší možné celkové ohodnocení. Formálně*

$$MST = \min_{\text{kostra } T \subseteq G} \left( \sum_{e \in E(T)} w(e) \right).$$

Jak brzy uvidíme, zadaný problém má velmi jednoduché algoritmické řešení.

### **Hladové algoritmy**

Asi nejjednodušším způsobem řešení různých optimalizačních úloh je “ber postupně to nejlepší, dokud se dá”. Tento postup se obecně v češtině nazývá *hladovým*, i když lepší by bylo použít správnější překlad anglického “greedy”, tedy *nenasytný*. A ještě výstižnější české slovo by bylo *postup “hamouna”*.

### Metoda 9.16. *Hladový postup*

Tato metoda řešení se používá u některých úloh, ve kterých hledáme co nejlepší řešení (optimalizujeme), skládající se z jednotlivých dílčích kroků (či elementů). Princip hladového postupu spočívá v tom, že ve vhodné pořadí bereme jednotlivé dílčí kroky úlohy a v každém se lokálně rozhodujeme pro nejlepší okamžitou možnost. Celkové řešení pak složíme z těchto dílčích nejlepších voleb.

**Značení:** Algoritmy implementující hladový postup se všeobecně nazývají *hladové algoritmy*.

**Poznámka, pozor!** Jistě jste se už setkali s tím, že postup hamouna obvykle nedává ty nejlepší výsledky (jak v teorii, tak ani v životě). Je to skutečně tak, hladový postup obvykle není matematicky korektní. Přesto však je třída optimalizačních úloh v diskrétní matematice, pro které hladový postup funguje výborně. Jedním příkladem fungování hladového postupu je právě hledání minimální kostry v grafu. Více příkladů bude uvedeno ve Cvičení 9.4.

Všeobecně lze říci, že hladový algoritmus dobře funguje všude tam, kde podstatu problému lze vyjádřit tzv. *matroidem*.

### Algoritmus 9.17. *Hladový pro minimální kostru.*

Je dán souvislý vážený graf  $G, w$  s nezáporným ohodnocením hran  $w$ .

- Seřadíme hrany grafu  $G$  vzestupně podle jejich ohodnocení, tj.  $w(e_1) \leq w(e_2) \leq \dots \leq w(e_m)$ .
- Začneme s prázdnou množinou hran  $T = \emptyset$  pro kostru.
- Pro  $i = 1, 2, \dots, m$  vezmeme hranu  $e_i$  a pokud  $T \cup \{e_i\}$  nevytváří kružnici, přidáme  $e_i$  do  $T$ . Jinak  $e_i$  “zahodíme”.
- Na konci množina  $T$  obsahuje hrany minimální kostry váženého grafu  $G, w$ .

**Důkaz správnosti Algoritmu 9.17:**

Nechť  $T$  je množina hran získaná v Algoritmu 9.17 a nechť hrany jsou již seřazené  $w(e_1) \leq w(e_2) \leq \dots \leq w(e_m)$ . Vezměme množinu hran  $T_0$  té minimální kostry (může jich být více se stejnou hodnotou), která se s  $T$  shoduje na co nejvíce prvních hranách. Pokud  $T_0 = T$ , algoritmus pracoval správně.

Předpokládejme tedy, že  $T_0 \neq T$ , a ukážeme spor, tj. že toto nemůže ve skutečnosti nastat. Označme  $j > 0$  takový index, že se množiny  $T_0$  a  $T$  shodují na prvních  $j - 1$  hranách  $e_1, \dots, e_{j-1}$ , ale neshodují se na  $e_j$ . To znamená, že  $e_j \in T$ , ale  $e_j \notin T_0$ . (Jistě nemůže nastat  $e_j \notin T$ , ale  $e_j \in T_0$ .) Podle Důsledku 9.5 obsahuje graf na hranách  $T_0 \cup \{e_j\}$  právě jednu kružnici  $C$ . Kružnice  $C$  však nemůže být obsažena v nalezené kostře  $T$ , a proto existuje hrana  $e_k$  v  $C$ , která  $e_k \notin T$  a zároveň  $k > j$ . Potom však je  $w(e_k) \geq w(e_j)$  podle našeho seřazení hran, a tudíž kostra na hranách  $(T_0 \setminus \{e_k\}) \cup \{e_j\}$  (vzniklá nahrazením hrany  $e_k$  hranou  $e_j$ ) není horší než  $T_0$  a měli jsme ji v naší úvaze zvolit místo  $T_0$ . To je hledaný spor.  $\square$

**Komentář:** Správný pohled na předchozí důkaz by měl být následovný: Předpokládali jsme, že nalezená kostra  $T$  se s některou optimální kostrou shoduje aspoň na prvních  $j - 1$  hranách. Poté jsme ukázali, že některou další hranu  $e_k$  v (předpokládané) optimální kostře lze zaměnit hranou  $e_j$ , a tudíž dosáhnout shodu aspoň na prvních  $j$  hranách. Dalšími iteracemi záměn ukážeme úplnou shodu naší nalezené kostry  $T$  s některou optimální kostrou. V našem důkaze jsme se vlastně zaměřili na důkaz toho, že ta poslední iterace záměn nemůže selhat. Nakreslete si tento důkaz obrázkem!

Základní hladový algoritmus pro hledání minimální kostry grafu byl poprvé explicitně popsán Kruskalem, ale už dříve byly objeveny jeho podobné varianty, které zde jen stručně zmíníme.

### Algoritmus 9.18. *Jarníkův pro minimální kostru.*

Hrany na začátku neseřazujeme, ale začneme kostru vytvářet z jednoho vrcholu a v každém kroku přidáme nejmenší z hran, které vedou z již vytvořeného podstromu do zbytku grafu.

**Poznámka:** Tento algoritmus je velmi vhodný pro praktické výpočty a je dodnes široce používaný. Málodko ve světě však ví, že pochází od Vojtěcha Jarníka, známého českého matematika — ve světové literatuře se obvykle připisuje Američanu Primovi, který jej objevil až skoro 30 let po Jarníkovi.

Avšak historicky vůbec první algoritmus pro problém minimální kostry (z roku 1928) byl nalezen jiným českým matematikem:

### Algoritmus 9.19. *Borůvkův pro minimální kostru.*

Toto je poněkud složitější algoritmus, chová se jako Jarníkův algoritmus spuštěný zároveň ze všech vrcholů grafu najednou. Detaily lze nalézt v doplňkové literatuře [5, Oddíl 4.4].

#### Úlohy k řešení

(9.4.1) Kolik koster má kružnice délky 2004?

\*(9.4.2) Zkuste sami vlastním počítáním odvodit, že počet koster úplného grafu  $K_5$  je  $125 = 5^3$ .

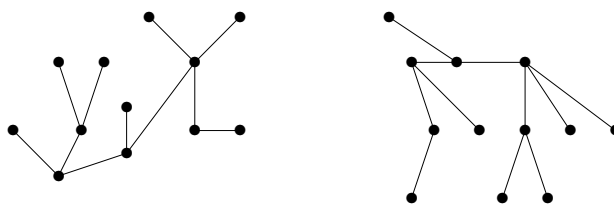
Návod: Zkuste něco chytřejšího, než kreslení všech koster. Co třeba se podívat na jednotlivé “typy” koster?

#### Rozšiřující studium

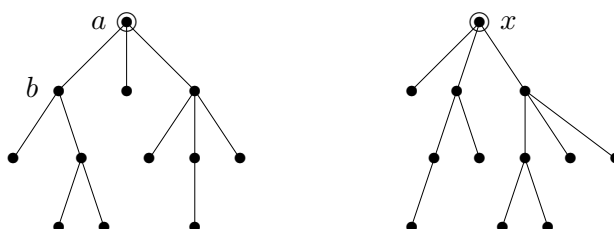
Stromům, jejich isomorfismu a kódování se věnují [5, Oddíly 4.1,2]. Problém minimální kostry a jeho historie jsou výborně shrnuty v [5, Oddíly 4.3,4,5]. Hladový postup je běžně zmiňován a používán v knihách zabývajících se algoritmy a optimalizací. Pro konkrétní algoritmické implementace odkazujeme i na [3]. Zájemci o hlubší studium počítání koster v grafech si mohou přečíst (poměrně obtížnou) [5, Kapitola 7].

## 9.5 Cvičení: Příklady o stromech, kostrách a hladovém postupu

**Příklad 9.20.** Zjistěte, zda následující dva stromy jsou isomorfní.



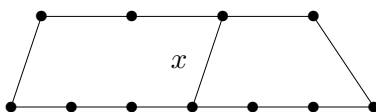
Nejprve si ověříme, že stromy mají stejný počet vrcholů (hran je pak také stejně). K daným dvěma stromům najdeme jejich centra a překreslíme si je jako kořenové stromy s kořeny v centrech.



Nyní přejdeme k určení minimálního závorkového kódu pro levý strom (Algoritmus 9.12): Například při rekurzivním volání ve vrcholu  $b$  určíme kódy jeho podstromů jako  $'()'$  a  $'(())'$ . Jelikož levou závorku považujeme za slovníkově menší, tyto dva podkódy seřadíme a spojíme takto  $'((())())'$ . Obdobně postupujeme dále... Nakonec v kořeni  $a$  získáme rekurzivními voláními kódy jeho tří podstromů  $'((())())'$ ,  $'()'$  a  $'((())())'$ . Po seřazení a spojení nám celkový minimální kód levého stromu vyjde  $'(((())())((())())())'$ .

Obdobně budeme postupovat při rekurzivním určování minimálního kódu pravého stromu. Zde nám podkódy jednotlivých tří podstromů kořene  $x$  vyjdou  $'()'$ ,  $'((())())'$  a  $'(((())())())'$ . Po seřazení a spojení nám celkový minimální kód pravého stromu vyjde  $'(((())())((())())())'$ . Napíšeme si tedy získané dva kódy pod sebe a uvidíme, kde se liší  $\begin{matrix} (((())())((())())()) \\ (((())())())((())())() \end{matrix}$ . Dané dva stromy tudíž nejsou isomorfní.  $\square$

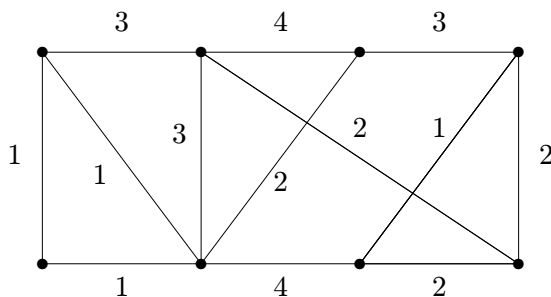
**Příklad 9.21.** Kolik různých koster má tento graf?



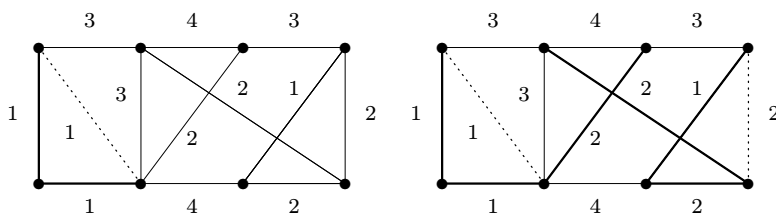
Jak vidíme, opět musíme vymazat dvě hrany z grafu, aby zbyla kostra. Nelze však vypouštět jednu hranu z levé kružnice a jednu z pravé, neboť by výběry nezyly nezávislé – hrana  $x$  je oběma sdílená. Podíváme se tedy na řešení jako na součet disjunktních možností, počtu koster obsahujících  $x$  a počtu koster bez hrany  $x$ .

Pokud hranu  $x$  vymažeme, zbude kružnice délky 11 a ta má 11 koster. Pokud naopak hranu  $x$  zachováme, musíme z levé kružnice od  $x$  vymazat jednu ze 6 hran a z pravé kružnice jednu z 5 hran. Tyto výběry již jsou nezávislé a počet možností je  $6 \cdot 5 = 30$ . Celkem má náš graf  $11 + 30 = 41$  koster.  $\square$

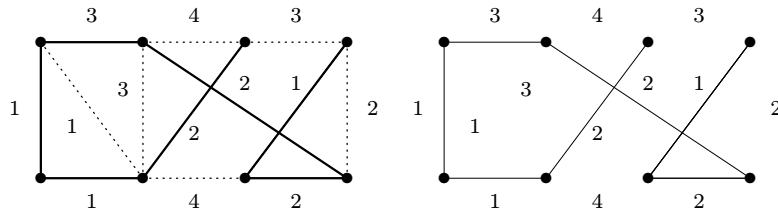
**Příklad 9.22.** Najděme hladovým algoritmem minimální kostru v tomto váženém grafu (váhy hran jsou zapsány čísly v obrázku).



Hrany si nejprve seřadíme podle jejich vah 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 4, 4. (na pořadí mezi hranami stejné váhy nezáleží, proto jej zvolíme libovolně). Začneme s prázdnou množinou hran (budoucí) kostry. Pak hladovým postupem přidáme první dvě hrany váhy 1 vlevo dole, které nevytvoří kružnici. Třetí hrana váhy 1 vlevo s nimi už tvoří trojúhelník, a proto ji přidat nelze, je zahozena. Při znázornění průběhu používáme tlusté čáry pro vybrané hrany kostry a tečkované čáry pro zahozené hrany:



Poté přejdeme na hrany s vahou 2, z nichž lze tři postupně přidat bez vytvoření kružnice a čtvrtá (úplně vpravo) již kružnici vytvoří a je proto zahozena. Viz. obrázek vpravo. Nakonec ještě přidáme hranu nejmenší vyšší váhy 3 vlevo nahoře a zbylé hrany již zahodíme, protože všechny tvoří kružnice.

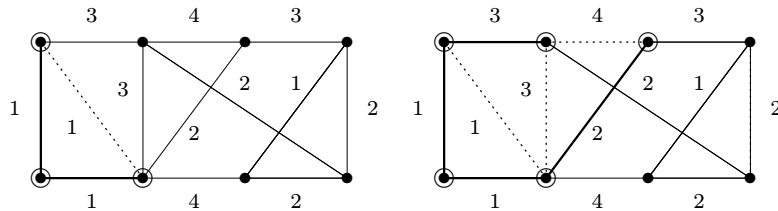


Získáme tak minimální kostru velikosti  $1 + 2 + 2 + 3 + 1 + 1 + 2 = 12$ , která je v tomto případě (náhodou) cestou, na posledním obrázku vpravo.

Poznamenáváme, že při jiném seřazení hran stejné váhy by kostra mohla vyjít jinak, ale vždy bude mít stejnou minimální velikost 12. (Například místo levé svislé hrany může obsahovat přilehlou úhlopříčku stejné váhy 1.)  $\square$

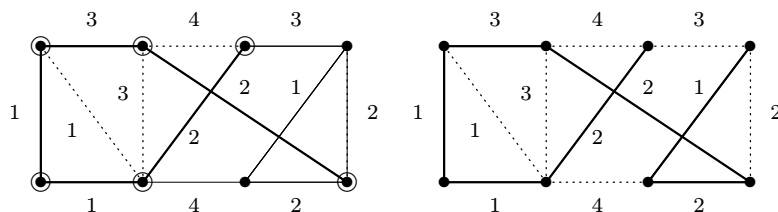
**Příklad 9.23.** Najděme minimální kostru grafu z Příkladu 9.22 Jarníkovým algoritmem.

Kostru začneme budovat v levém dolním vrcholu. (Tj. naše částečná kostra zatím dosáhla jen na levý dolní vrchol.) Vidíme, že obě hrany z něj vycházející mají váhu 1, proto je obě do kostry přidáme. Takto naše budovaná kostra již dosáhla na tři vrcholy grafu, které si v následujícím obrázku vlevo vyznačíme. (Zbylé hrany mezi dosaženými vrcholy jsou již k ničemu, proto je zahodíme.)



V dalším kroku ze všech tří dosažených vrcholů vybíráme nejmenší hranu vedoucí mimo ně. Jak hned vidíme, nejmenší taková hrana má váhu 2, takže ji k naší kostře přidáme a dosáhneme čtvrtý vrchol grafu. Poté opět vybíráme nejmenší hranu z dosažených vrcholů ven, ale ty mají nyní váhu nejméně 3 (zvolíme tu nahoře vlevo). Také ji přidáme do kostry a dostaneme se do situace vyznačené na horním obrázku vpravo.

Nyní již je dosažených vrcholů 5 a nejmenší z hran vedoucích z dosažených ven má váhu 2. (Všimněte si dobře tohoto rozdílu od základního hladového postupu – v tomto algoritmu se nemusí hrany probírat globálně monotónně podle svých vah!) Takto dosáhneme i pravého dolního vrcholu, na následujícím obrázku vlevo.



Nakonec ještě dosáhneme zbylé dva vrcholy hranami po řadě vah 2 a 1. Získáme tak stejnou minimální kostru jako v Příkladě 9.22. Opět je však možnost nalézt jinou z

minimálních koster stejné velikosti, pokud mezi hranami stejné váhy vedoucími ven v každém kroku vybíráme jinak.  $\square$

V dalších dvou příkladech si ukážeme použití hladového postupu na řešení jiných problémů než minimální kostry. Obě aplikace postupu jsou podobné a poměrně přirozené, přesto jen první z nich je matematicky korektní – dává vždy optimální výsledek, kdežto druhá ne.

**Příklad 9.24.** Podívejme se na tento přirozený problém z univerzitního prostředí: Kroužky studentů mají během dne pevně naplánované časy cvičení. Naším úkolem je rozmístit cvičení do co nejméně učeben, aby se samozřejmě v jedné učebně cvičení nepřekrývala. K vyřešení použijeme hladový postup:

- Nejdříve začínající cvičení umístíme do učebny č. 1.
- Vybereme nejdříve začínající neumístěné cvičení (libovolné, pokud jich více začíná ve stejnou dobu) a umístíme jej do první volné učebny nejmenšího čísla.
- Opakujeme předchozí bod, dokud nejsou všechna cvičení rozmístěna.

Jak dobrý výsledek (tj. maximální počet potřebných učeben) nám tento postup dá?

Je to možná s podivem, ale tento primitivní postup nám dá zcela optimální řešení úlohy, jak si nyní stručně zdůvodníme:

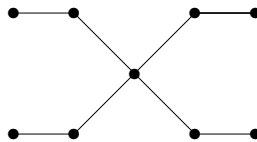
Předpokládejme, že učebny jsou číslovány  $1, 2, 3, \dots$ . Pokud uvedený hladový postup potřebuje celkem  $k$  učeben, znamená to, že v některém svém kroku již měl učebny  $1, 2, \dots, k-1$  obsazené probíhajícími cvičeními a pro další cvičení musel použít učebnu číslo  $k$ . (Jinak by použil učebnu menšího čísla.) To však znamená, že v onom okamžiku se najednou koná  $k$  různých cvičení, a proto použití nejméně  $k$  učeben je nutné.  $\square$

**Příklad 9.25.** Vrcholovým pokrytím v grafu  $G$  nazveme množinu vrcholů  $X \subseteq V(G)$  takovou, že každá hrana grafu  $G$  má aspoň jeden konec v  $X$ . Zadaným úkolem je nalézt v grafu vrcholové pokrytí nejmenší možné velikosti. Použijeme hladový postup (kdy se snažíme každým krokem pokrýt co nejvíce ze zbylých hran v grafu):

- Začneme s prázdnou  $X = \emptyset$ .
- Vybereme vrchol  $v$  největšího stupně v  $G$  a přidáme jej do  $X$ .
- Vrchol  $v$  odebereme z grafu  $G$  (i s jeho hranami) a jdeme zpět na předchozí bod, dokud nezbude graf  $G$  prázdný.

Proč je toto řešení obecně nekorektní?

Popsaný hladový postup v některých případech nenalezne nejmenší možnou množinu vrcholového pokrytí. Podívejme se na tento graf:



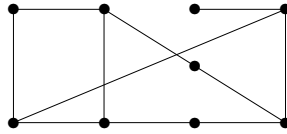
Hladový postup nejprve vybere prostřední vrchol, ale pak ještě musí v dalších čtyřech krocích vybrat 4 vrcholy, tedy celkem vyjde množina  $X$  velikosti 5:



Snadno však zjistíme, že optimální vrcholové pokrytí má velikost jen 4 a je vyznačené na obrázku vpravo. □

### Úlohy k řešení

(9.5.1) Kolik hran je třeba vypustit z následujícího grafu na 9 vrcholech, aby zbyla jeho kostra?



(9.5.2) Kolik vrcholů má strom s 2004 hranami? Je to jednoznačné?

(9.5.3) Les má 2009 vrcholů a celkem 4 souvislé komponenty. Kolik má hran?

(9.5.4) Kolik komponent souvislosti má les s 2004 vrcholy a 1993 hranami?

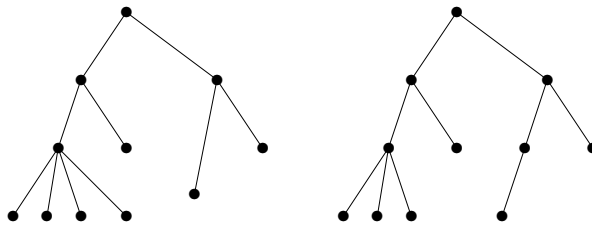
(9.5.5) Mějme čísla  $\{10, 11, 13, 14, 15, 21\}$  a spojme dvojice z nich hranami, pokud jsou soudělná. Je výsledný graf stromem? A aspoň lesem?

(9.5.6) Které z následujících výrazů jsou správnými závorkovými kódy pro nakreslené uspořádané kořenové stromy?

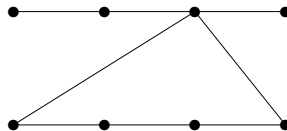
A: (((() () ()) ()) ((()) ()))

B: (((() () () ()) ()) ((()) ()))

C: (((() () ()) ()) ((() ())))



(9.5.7) Kolik různých koster má tento graf?

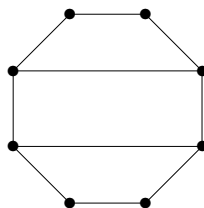


(9.5.8) Kolik koster může mít graf, který vznikne ze stromu na 8 vrcholech přidáním libovolné nové hrany (mezi dvěma nespojenými vrcholy)? Uvědomte si, že pro různé stromy a hrany mohou vyjít různé výsledky, a vaším úkolem je najít všechny možné počty, včetně příslušných obrázků.

(9.5.9) Kolik hran je třeba vypustit z úplného grafu na  $n$  vrcholech, aby vznikla jeho kostra?

\*(9.5.10) Kolik nejméně vrcholů musí mít graf (bez násobných hran), ve kterém lze nalézt dvě kostry nesdílející žádnou hranu? Zdůvodněte.

\*(9.5.11) Kolik koster má graf na následujícím obrázku?

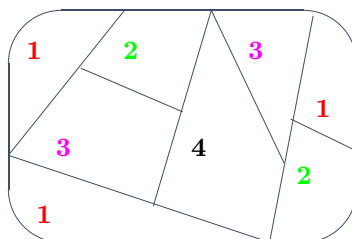


## 10 Barevnost a kreslení grafů

### Úvod

Již dříve jsme si připomínali jeden z historických kamenů teorie grafů – slavný problém sedmi mostů v Královci (dnešním Kaliningradě). Další neméně slavný problém pochází z poloviny 19. století a je obvykle zvaný problémem čtyř barev. Na rozdíl od sedmi mostů, problém čtyř barev zůstal nevyřešený po více než 100 let! A právě snahy o jeho vyřešení se zapříčinily o rozvoj mnoha oblastí teorie grafů.

Problém čtyř barev byl původně formulován pro politické mapy států: Výrobci map chtěli státy barevně odlišit, aby každé dva sousední měli různé barvy. Přitom chtěli mapy tisknout co nejlevněji, tedy s nejmenším možným počtem barev. Není problém nakreslit čtyři státy tak, že každý s každým sousedí, a proto 4 barvy jsou někdy potřebné. Pro větší příklad se podívejme na následující obarvenou “mapu”:



Tiskařská praxe brzy ukázala, že 4 barvy vždy stačí, ale matematici si dlouho lámali hlavy s tím, proč tomu tak je. . .

Zde si ukážeme, jak uvedený problém souvisí s grafy a hlavně s pojmy rovinného kreslení a barevnosti grafů. Uvedeme se (a zčásti odvodíme) základní vlastnosti rovinných grafů a grafové barevnosti.

### Cíle

Prvním cílem této lekce je definovat barevnost grafů a naučit se s ní pracovat. Druhým cílem je definovat rovinné grafy a přehledově ukázat jejich vlastnosti, především ve vztahu k jejich barevnosti a ke geometrii.

### 10.1 Barevnost grafu

Nejprve si formálně definujeme pojem barevnosti – představme si, že hrany grafu nám říkají, že jejich koncové vrcholy musí být barevně odlišené (třeba proto, že reprezentují sousední státy, nebo proto, že jinak jsou si příliš podobné, atd). Samozřejmě bychom mohli každému vrcholu grafu dát jinou barvu, ale k čemu by pak takový problém byl? My bychom chtěli použít barev celkem co nejméně.

**Definice:** *Obarvením grafu*  $G$  pomocí  $k$  barev myslíme libovolné zobrazení

$$c : V(G) \rightarrow \{1, 2, \dots, k\}$$

takové, že každé dva vrcholy spojené hranou dostanou různé barvy, tj.  $c(u) \neq c(v)$  pro všechny  $\{u, v\} \in E(G)$ .

**Definice 10.1.** *Barevnost* grafu  $G$

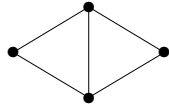
je nejmenší přirozené číslo  $\chi(G)$  pro které existuje obarvení grafu  $G$  pomocí  $\chi(G)$  barev.

Čísla  $1, 2, \dots, k$  z předchozí definice tak nazýváme barvami vrcholů (je to pohodlnější, než popisovat barvy běžnými jmény jako bílá, červená, atd).

**Poznámka:** Uvědomme si, že barevnost lze definovat pouze pro graf bez smyček, protože oba konce smyčky mají vždy stejnou barvu a nic s tím nenaděláme.



**Příklad 10.2.** Určete barevnost grafu



Na první pohled je vidět, že potřebujeme aspoň 3 barvy, neboť graf má trojici navzájem spojených vrcholů (trojúhelník). Na druhý pohled pak zjistíme, že levý a pravý vrchol spojené hranou nejsou, a proto jim lze přiřadit stejnou barvu (a další dvě barvy vrchnímu a spodnímu vrcholu). Proto má graf barevnost 3.  $\square$

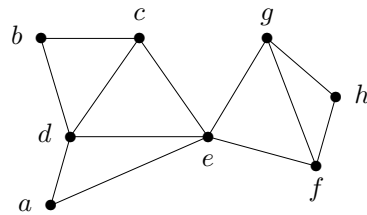
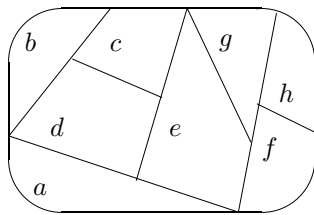
V obecnosti lze o barevnosti grafů říci následující.

**Lema 10.3.** Necht  $G$  je jednoduchý graf (bez smyček) na  $n$  vrcholech. Pak  $\chi(G) \leq n$  a rovnost nastává právě když  $G \simeq K_n$  je úplný graf.

**Důkaz:** Stačí každý vrchol obarvit jinou barvou a máme skutečné obarvení  $n$  barvami dle definice. Navíc pokud některá dvojice  $u, v$  vrcholů není spojená hranou, můžeme volit lepší obarvení  $c(u) = c(v) = 1$  a zbylé vrcholy různými barvami  $2, 3, \dots, n - 1$ , tj. pak  $\chi(G) < n$ .  $\square$

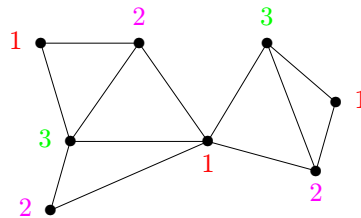
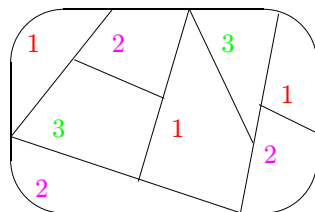
**Příklad 10.4.** Vraťme se k příkladu barvení mapy z úvodu lekce a ukažme si, jak souvisí s grafy a jejich barevností.

Označme si státy na mapě písmeny  $a, b, \dots, h$ .



Jednotlivé oblasti na mapě (předpokládáme, že každý stát má souvislé území, tj. státy = oblasti) prohlásíme za vrcholy našeho grafu a sousední dvojice států spojíme hranami. Nezapomeňme přitom, že “sousední” znamená sdílení celého úseku hranice, ne jen jednoho rohu. Výsledkem bude graf nakreslený vpravo. Podíváme-li se nyní zpátky na definici barevnosti grafu, vidíme, že se jedná přesně o ten samý problém jako u barvení původní mapy.

Při troše snahy také najdeme lepší obarvení uvedené mapy využívající pouhých tří barev:



$\square$

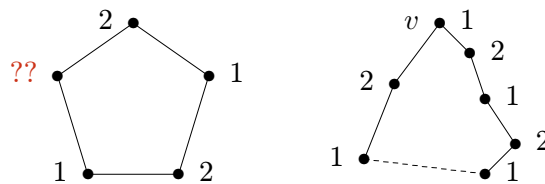
Podívejme se, které grafy mají nízkou barevnost. Je jasné, že jedna barva stačí, jen pokud graf nemá hrany. Grafy obarvitelné dvěma barvami už tvoří zajímavější třídu a jsou obvykle nazývané jedním slovem *bipartitní*. Poměrně jednoduchý popis bipartitních grafů je uvedený v následujícím tvrzení.

**Věta 10.5.** Graf  $G$  má barevnost 1 právě když nemá žádné hrany. Graf  $G$  má barevnost 2 právě když nemá žádnou kružnici liché délky jako podgraf.

**Důkaz:** Pokud graf nemá hrany, můžeme všechny vrcholy obarvit stejnou barvou 1. Naopak pokud mají všechny vrcholy stejnou barvu, nemůže graf mít žádnou hranu.

Druhá část je už trochu složitější, proto důkaz jen naznačíme. Na jednu stranu, lichou kružnici nelze obarvit dvěma barvami, viz obrázek. Na druhou stranu si představme, že zvolíme libovolný vrchol  $v$  grafu  $G$  s barvou 1 a ostatní vrcholy obarvíme takto: Vrcholy, jejichž vzdálenost od  $v$  je lichá, obarvíme 2. Vrcholy, jejichž vzdálenost od  $v$  je sudá, obarvíme 1.

Pokud bychom tak získali třeba dva vrcholy v sudé vzdálenosti spojené hranou, nalezneme při troše snahy lichou kružnici procházející touto hranou, což nelze. Stejně tak pro dva vrcholy v liché vzdálenosti. Proto naše obarvení za daných předpokladů nemůže dát stejnou barvu sousedním vrcholům, a tudíž dvě barvy stačí.

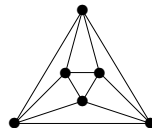


□

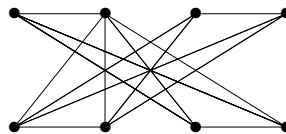
Poněkud nepříjemnou skutečností je, že o barvení a určování barevnosti grafů nemůžeme v obecnosti říci o mnoho více, než jsme uvedli výše. Jedná se o problém algoritmicky ještě obtížnější než byl problém isomorfismu a nepředpokládá se, že by se barevnost grafu dala algoritmicky určit jinak, než hrubou silou probráním (téměř) všech možných obarvení.

### Úlohy k řešení

(10.1.1) Kolika nejméně barvami korektně obarvíte graf pravidelného osmistěnu?



(10.1.2) Kolik nejméně barev je třeba na korektní obarvení tohoto grafu? Příslušné obarvení vyznačte.



(10.1.3) Jaká je barevnost stromu?

(10.1.4) Kolik barev vždy stačí na korektní obarvení grafu vzniklého z kružnice délky 2004 přidáním jedné nové hrany?

\*(10.1.5) Představme si, že nějaký velký graf má všechny vrcholy stupně nejvýše 101. Kolika barvami takový graf jistě obarvíme?

## 10.2 Rovinné kreslení grafu

Dalším důležitým grafovým pojmem úzce svázaným s problémem čtyř barev je rovinné nakreslení, tj. nakreslení bez překřížených hran. Jen málokteré grafy rovinné nakreslení

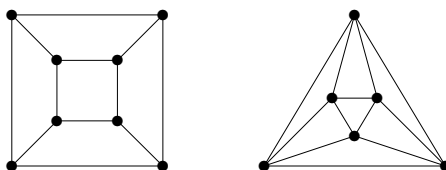
mají, ale důležitost grafů s rovinným nakreslením je motivována jak esteticky (nakreslení bez křížení hran vypadají hezky a jsou snadno “čitelná”), tak jejich teoretickým významem i praktickými aplikacemi (představme si jednostranný plošný spoj jako graf obvodu – hrany při jeho realizaci fyzicky nemůžeme křížit bez drátových propojek).

**Definice 10.6.** *Rovinným nakreslením* grafu  $G$

myslíme zobrazení, ve kterém jsou vrcholy znázorněny jako různé body v rovině a hrany jako oblouky (čili křivky) spojující body svých koncových vrcholů. Přitom hrany se nesmí nikde křížit ani procházet jinými vrcholy než svými koncovými body.

Graf je *rovinný* pokud má rovinné nakreslení.

**Komentář:** Důležitým příkladem rovinných grafů jsou grafy (třírozměrných Euklidovských) mnohostěnů, třeba graf čtyřstěnu, krychle, osmistěnu, dvanáctistěnu, atd.

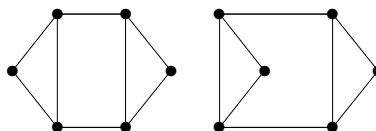


Platí, že grafy mnohostěnů jsou vždy rovinné a 3-souvislé. Naopak každý rovinný 3-souvislý jednoduchý graf je grafem nějakého mnohostěnu. (Důkaz tohoto tvrzení je obtížný.)

Geometrický příklad grafů mnohostěnů také přináší část terminologie rovinných kreslení a hlavně motivuje důležitý slavný Eulerův vztah (Věta 10.7).

**Definice:** *Stěnamí* rovinného nakreslení grafu nazýváme souvislé oblasti roviny ohraničené tímto nakreslením grafu.

**Komentář:**

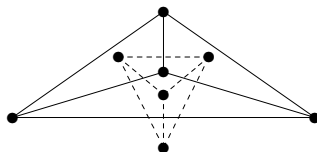


Rovinný graf může mít více *podstatně různých* nakreslení, jak vidíme na uvedeném ilustračním obrázku, ale platí, že 3-souvislý rovinný graf má ve všech svých rovinných nakresleních stejné stěny. To intuitivně znamená, že ze znalosti grafu mnohostěnu můžeme odvodit přibližný tvar původního tělesa.

Podívejte se zpět na konstrukci použitou v Příkladě 10.4 – oblasti, neboli stěny, mapy jsme nahrazovali vrcholy nového grafu a spojovali jsme hranami sousedící dvojice oblastí. Tuto konstrukci lze formálně zobecnit na stěny nakreslení libovolného grafu:

**Definice.** *Duální graf* rovinného nakreslení grafu  $G$  získáme tak, že stěny nahradíme vrcholy duálu a hranami spojíme sousedící dvojice stěn.

**Komentář:** Duální graf k rovinnému grafu je vždy rovinný, což je snadné dokázat. Pro příklad odvození duálního grafu se podívejme na obrázek:



Nyní si uvedeme zajímavý a vlastně “jediný rozumný” kvantitativní vztah o rovinných nakresleních grafů. Jedná se o slavný **Eulerův vztah**, který říká:

**Věta 10.7.** *Nechť rovinné nakreslení neprázdného souvislého grafu  $G$  má  $f$  stěn. Pak*

$$|V(G)| + f - |E(G)| = 2.$$

**Důkaz:** Nechť počet vrcholů v  $G$  je  $v$  a hran  $h$ . Důkaz této důležité věty pouze naznačíme, detaily lze najít v literatuře.

- Pokud je  $G$  strom, tj. nemá kružnice, má ve svém nakreslení jedinou stěnu a dle Věty 9.3 má přesně  $h = v - 1$  hran. Potom platí  $v + f - h = v + 1 - (v - 1) = 2$ .
- Pokud  $G$  obsahuje kružnici  $C$ , pak vypustíme jednu její hranu  $e$ . Tím se počet hran sníží o 1, ale zároveň se sníží o 1 počet stěn, protože kružnice  $C$  původně oddělovala dvě stěny přilehlé k hraně  $e$  od sebe, ale nyní tyto dvě stěny splynou v jednu. Počet vrcholů se nezmění. Proto se nezmění hodnota  $v + f - h = v + (f - 1) - (e - 1) = 2$ .

Tvrzení tak plyne z principu matematické indukce. □

**Poznámka:** Všimněte si dobře, že Eulerův vztah vůbec nezávisí na tom, jak je graf  $G$  nakreslený, je to vlastnost grafu jako takového.

Tento jednoduše vypadající vztah má mnoho aplikací a důsledků, z nichž část si uvedeme i dokážeme.

**Důsledek 10.8.** *Jednoduchý rovinný graf na  $v \geq 3$  vrcholech má nejvýše  $3v - 6$  hran. Jednoduchý rovinný graf na  $v \geq 3$  vrcholech a bez trojúhelníků má nejvýše  $2v - 4$  hran.*

**Důkaz:** Můžeme předpokládat, že graf je souvislý, jinak bychom přidali další hrany. Nechť počet vrcholů v  $G$  je  $v$ , stěn je  $f$  a hran  $h$ . Jelikož nemáme smyčky ani násobné hrany, má každá stěna v nakreslení grafu na obvodu aspoň 3 hrany, přitom každou hranu započítáme ve dvou přilehlých stěnách. Pak tedy platí  $e \geq \frac{1}{2} \cdot 3f$ , neboli  $\frac{2}{3}e \geq f$ . Dosazením do vztahu Věty 10.7 získáme

$$2 = v + f - e \leq v + \frac{2}{3}e - e = v - \frac{1}{3}e$$

$$e \leq 3(v - 2) = 3v - 6.$$

Druhá část se dokazuje obdobně, ale nyní víme, že graf nemá ani trojúhelníky, a tudíž má každá stěna v nakreslení grafu na obvodu aspoň 4 hrany. Pak tedy platí  $e \geq \frac{1}{2} \cdot 4f$ , neboli  $\frac{2}{4}e \geq f$ . Dosazením do vztahu Věty 10.7 získáme

$$2 = v + f - e \leq v + \frac{2}{4}e - e = v - \frac{1}{2}e$$

$$e \leq 2(v - 2) = 2v - 4.$$

Tím jsme hotovi. □

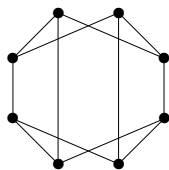
**Důsledek 10.9.** *Každý rovinný graf obsahuje vrchol stupně nejvýše 5. Každý rovinný graf bez trojúhelníků obsahuje vrchol stupně nejvýše 3.*

**Důkaz:** Pokud by všechny vrcholy měly stupně alespoň 6, celý graf by měl aspoň  $\frac{1}{2} \cdot 6v = 3v$  hran, což je ve sporu s Důsledkem 10.8. Některý vrchol musí tudíž mít menší stupeň než 6.

Obdobně postupujeme u druhého tvrzení. □

## Úlohy k řešení

(10.2.1) Nakreslete rovinně tento graf:



(10.2.2) Graf pravidelného dvacetistěnu má 12 vrcholů a 20 stěn. Kolik má hran?

(10.2.3) Co je duálním grafem k rovinnému nakreslení grafu krychle?

(10.2.4) K rovinnému nakreslení stromu přidáme dvě nekřížící se hrany. Kolik bude mít výsledný graf stěn?

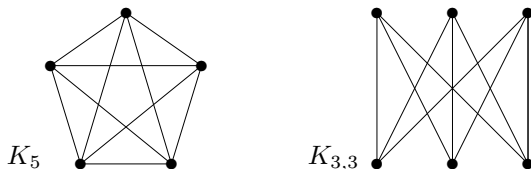
## 10.3 Rozpoznání rovinných grafů

Při praktickém využití rovinných grafů je potřeba umět abstraktně zadaný graf rovinně nakreslit bez křížení hran. Na rozdíl od problému určení barevnosti grafu se našťastí jedná o efektivně algoritmicky řešitelný problém.

**Fakt.** Existují algoritmy, které pro zadaný graf efektivně (rychle) rozhodnou, zda je to rovinný graf, a případně naleznou rovinné nakreslení. Tyto algoritmy však nejsou jednoduché.

Místo obecných algoritmů pro rovinné kreslení grafů se zde podíváme na otázku, jak odůvodnit nerovinnost (malého) grafu.

**Příklad 10.10.** Dokažme, že následující dva grafy,  $K_5$  a  $K_{3,3}$  nejsou rovinné.



Při zdůvodnění využijeme znalosti předchozího oddílu. Všimněme si, že graf  $K_5$  má 5 vrcholů a  $10 > 3 \cdot 5 - 6$  hran. Podobně graf  $K_{3,3}$  má 6 vrcholů a  $9 > 2 \cdot 6 - 4$  hran, přitom neobsahuje žádné trojúhelníky. Proto podle Důsledku 10.8 žádný z nich není rovinný. (Pokud by byl rovinný, počet jeho hran by musel být menší, než skutečně je.)  $\square$

**Důsledek 10.11.** Grafy  $K_5$  a  $K_{3,3}$  nejsou rovinné.

Význam grafů  $K_5$  a  $K_{3,3}$  uvedených v předchozím důsledku spočívá v tom, že jsou to “nejmenší” nerovinné grafy ve velmi silném významu slova nejmenší – každý další nerovinný graf jeden z nich “obsahuje”. Pro uvedení takového popisu rovinných grafů ještě potřebujeme jednu definici.

**Definice:** *Podrozdělením* grafu  $G$  rozumíme graf, který vznikne z  $G$  rozdělením některých hran novými vrcholy stupně 2.



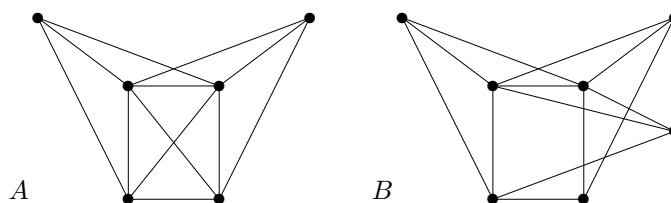
Důležitý abstraktní popis všech rovinných grafů našel K.Kuratowski:

**Věta 10.12.** *Graf  $G$  je rovinný právě když neobsahuje podrozdělení grafů  $K_5$  nebo  $K_{3,3}$  jako podgrafy.*

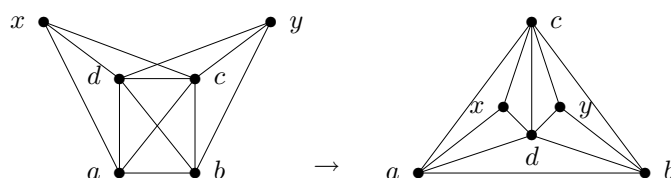
**Poznámka** o rozpoznání nerovinnosti grafu: Pokud chceme ukázat, že daný graf je rovinný, prostě jej nakreslíme v rovině bez křížení hran. Předchozí věta nám naopak dává spolehlivý způsob, jak ukázat, že daný graf není rovinný – prostě v něm najdeme podrozdělení grafů  $K_5$  nebo  $K_{3,3}$ . (Ve skutečnosti tuto obtížnou větu ani nepotřebujeme ke zdůvodnění nerovinnosti, stačí nám Důsledek 10.11. Věta 10.12 nám jen říká, že příslušná podrozdělení vždy v nerovinných grafech najdeme.)

Pro praktické použití věty dodáme, že až na vzácné výjimky se lépe v nerovinných grafech najde podrozdělení grafu  $K_{3,3}$  než grafu  $K_5$ .

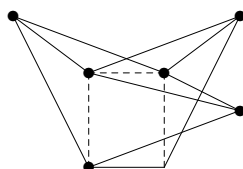
**Příklad 10.13.** *Které z následujících dvou grafů jsou rovinné? Najděte rovinné nakreslení (včetně očíslovaných vrcholů), nebo zdůvodněte nerovinnost grafu.*



Po chvíli zkoumání určitě přijdeme na to, že graf  $A$  se dá nakreslit rovinně takto:



Graf  $B$  na druhou stranu rovinný není podle Věty 10.12, protože je v něm obsaženo podrozdělení grafu  $K_{3,3}$ , které je ukázáno na tomto obrázku:



□

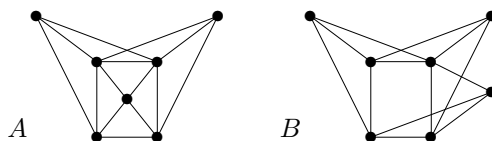
Závěrem si ještě bez důkazu uvedeme, že rovinné grafy vždy mají “pěkné” nakreslení v rovině.

**Věta 10.14.** *Každý jednoduchý rovinný graf lze nakreslit v rovině (bez křížení hran) tak, že hrany jsou úsečky.*

### Úlohy k řešení

(10.3.1) *Pro která  $n$  je úplný graf  $K_n$  rovinný?*

(10.3.2) *Které z následujících dvou grafů jsou rovinné? Najděte rovinné nakreslení (včetně očíslovaných vrcholů), nebo zdůvodněte nerovinnost grafu.*



(10.3.3) *Kolik hran stačí přidat ke kružnici, aby vznikl nerovinný graf?*

## 10.4 Barvení map a rovinných grafů

**Komentář:** Vzpomeňme si na již zmiňovaný převod mapy na graf – jedná se vlastně o vytvoření duálního grafu k této mapě (strana 86). Aby v duálním grafu k mapě nevznikly smyčky, v mapě nesmí žádný stát sousedit sám se sebou, což je přirozený požadavek.

V roce 1976 Appel a Haken, a pak v roce 1993 znovu Robertson, Seymour, Sanders a Thomas, dokázali tuto větu, která rozřešila problém čtyř barev a která je jedním z nejslavnějších výsledků diskrétní matematiky vůbec:

**Věta 10.15.** *Každý rovinný graf bez smyček lze obarvit 4 barvami.*

Důkaz této věty je nesmírně složitý (však byl také hledán po více než 100 let a k jeho úplnému provedení je stále třeba výkonný počítač), a proto si uvedeme slabší a mnohem jednodušší tvrzení:

**Tvrzení 10.16.** *Každý rovinný graf bez smyček lze obarvit 6 barvami.*

*Každý rovinný graf bez smyček a bez trojúhelníků lze obarvit 4 barvami.*

**Důkaz:** Postupujeme indukcí dle počtu vrcholů grafu  $G$ . Podle Důsledku 10.9 najdeme vrchol  $v$  v  $G$  stupně nejvýše 5. Podle indukčního předpokladu lze menší graf  $G-v$  obarvit 6 barvami. Z nich jen nejvýše 5 je sousedy vrcholu  $v$  a tudíž šestou barvou můžeme  $v$  dobarvit.

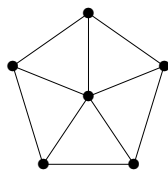
Druhou část dokážeme obdobně, když vezmeme vrchol stupně  $\leq 3$  v  $G$ . □

### Rozšiřující studium

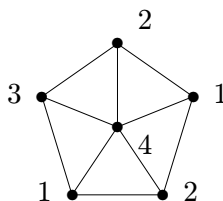
Problematika rovinného kreslení grafů a jejich barvení je pokryta v [5, Kapitola 5]. Zájemce o jednoduchý popis algoritmu na rovinné kreslení odkazujeme na [3]. Zájemci o podrobný popis historie problému čtyř barev zase mohou mnoho zajímavých informací nalézt na internetu.

## 10.5 Cvičení: Příklady na barevnost a rovinnost grafů

**Příklad 10.17.** *Určete a zdůvodněte barevnost tohoto grafu:*



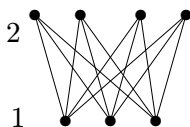
Vidíme, že obvodová kružnice tohoto grafu má délku 5, což je liché číslo, a proto je potřeba na její korektní obarvení použít aspoň tři barvy 1, 2, 3. Pak je ale středový vrchol spojený s každou z těchto barev 1, 2, 3, tudíž pro něj potřebujeme čtvrtou barvu 4.



To znamená, že daný graf má barevnost 4. □

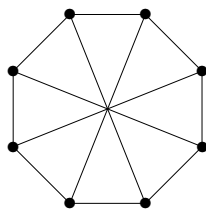
**Příklad 10.18.** Kolik nejméně hran musíte vypustit z úplného grafu na 7 vrcholech, aby se výsledný graf dal korektně obarvit dvěma barvami?

Musíme vypustit všechny hrany mezi vrcholy, kterým chceme dát stejnou barvu, aby výsledné obarvení bylo korektní. Jelikož všechny vrcholy úplného grafu jsou si ekvivalentní, stačí se rozhodnout, kolik z nich dostane barvu 1 a kolik barvu 2. Jak snadno zjistíme, optimální je barvy rozdělit napůl, tj. 3 a 4 v tomto případě. Je tedy potřeba vypustit nejméně  $\binom{3}{2} + \binom{4}{2} = 3 + 6 = 9$  hran.

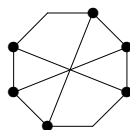


□

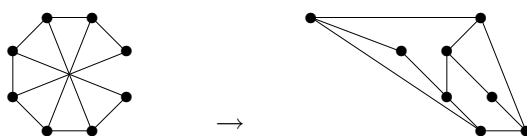
**Příklad 10.19.** Kolik nejméně hran je třeba vypustit z následujícího 8-vrcholového grafu, aby vznikl rovinný graf? Zdůvodněte.



Nejprve se podíváme, zda náš graf náhodou není rovinný. To ale není, protože zde vidíme v něm obsažené podrozdělení  $K_{3,3}$ :



Tento obrázek zároveň ukazuje ještě jednu zajímavost – náš graf nebude rovinný, ani když vypustíme libovolnou z “tětiv” obvodové kružnice. Na druhou stranu není těžké objevit, že stačí vypustit třeba pravou svislou hranu a získáme rovinné nakreslení:

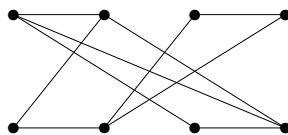


Stačí tedy vypustit jednu hranu a získáme rovinný graf.

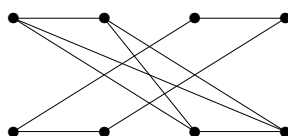
□

### Úlohy k řešení

**(10.5.1)** Kolik nejméně barev je třeba na korektní obarvení tohoto grafu? Příslušné obarvení vyznačte.

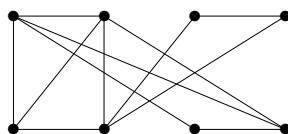


**(10.5.2)** Kolik nejméně barev je třeba na korektní obarvení tohoto grafu? Příslušné obarvení vyznačte.

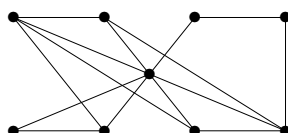




(10.5.3) Obarvěte následující graf na 8 vrcholech třemi barvami 1, 2, 3 tak, aby se barva 3 použila co nejvíce krát.

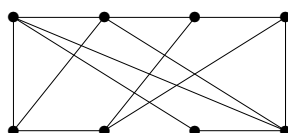


(10.5.4) Obarvěte následující graf na 9 vrcholech třemi barvami 1, 2, 3 tak, aby se barva 3 použila co nejméně krát.

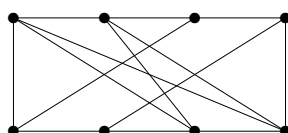


\*(10.5.5) Jaká může být barevnost grafu, který vznikne z úplného grafu na 15 vrcholech vypuštěním tří hran? (Je to jednoznačné?)

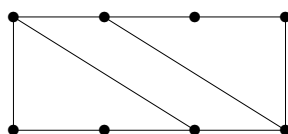
(10.5.6) Je tento graf rovinný?



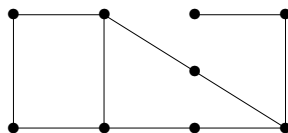
(10.5.7) Je tento graf rovinný?



(10.5.8) Kolik nejvýše hran lze přidat do následujícího grafu na 8 vrcholech, aby ještě zůstal rovinný a jednoduchý?



(10.5.9) Kolik nejvýše hran lze přidat do následujícího grafu na 9 vrcholech, aby ještě zůstal rovinný a jednoduchý?



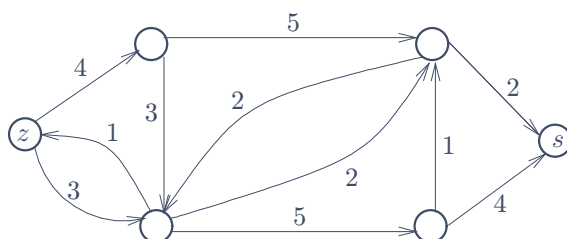
(10.5.10) Nakreslete libovolný rovinný graf s 18 hranami a 10 stěnami. Kolik má takový graf vrcholů?

## 11 Toky v sítích

### Úvod

Závěrem se podíváme ještě na jednu oblast úloh, kde našla teorie grafů bohaté uplatnění, konkrétně orientované grafy. Jde o oblast tzv. „síťových“ úloh: Pojem síť používáme jako souhrnné pojmenování pro matematické modely situací, ve kterých přepravujeme nějakou substanci (hmotnou či nehmotnou) po předem daných přepravních cestách, které navíc mají omezenou kapacitu.

Jedná se třeba o potrubní síť přepravující vodu nebo plyn, o dopravní síť silnic s přepravou zboží, nebo třeba o internet přenášející data. Obvykle nás zajímá problém přenést z daného „zdroje“ do daného cíle čili „stoku“ co nejvíce této substance, za omezujících podmínek kapacit jednotlivých přepravních cest (případně i jejich uzlů). Obrázkem můžeme vyjádřit síť s danými kapacitami přepravy jako:



Problém maximálního toku v síti je snadno algoritmicky řešitelný, jak si také popíšeme. Jeho řešení má (kupodivu) i mnoho teoretických důsledků, třeba pro souvislost či párování v grafech.

### Cíle

Úkolem této lekce je teoreticky popsat problém toku v síti a vysvětlit základní algoritmus nenasyčených cest pro jeho řešení. Dále jsou uvedeny některé důsledky vysvětlené látky (pro rozšířené síť, pro bipartitní párování a výběr reprezentantů množin).

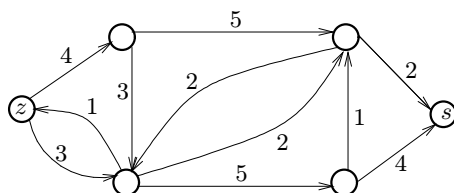
### 11.1 Definice sítě

Základní strukturou pro reprezentaci sítí je orientovaný graf. Vrcholy grafu modelují jednotlivé uzly sítě a hrany jejich spojnice. Vzpomeňme si (strana 48), že v orientovaných grafech je každá hrana tvořena uspořádanou dvojicí  $(u, v)$  vrcholů grafu, a tudíž taková hrana má směr z vrcholu  $u$  do  $v$ .

**Definice 11.1.** *Síť* je čtveřice  $S = (G, z, s, w)$ , kde

- $G$  je orientovaný graf,
- vrcholy  $z \in V(G)$ ,  $s \in V(G)$  jsou *zdroj* a *stok*,
- $w : E(G) \rightarrow \mathbb{R}^+$  je kladné ohodnocení hran, zvané *kapacita hran*.

Komentář:



Na obrázku je zakreslena síť s vyznačeným zdrojem  $z$  a stokem  $s$ , jejíž kapacity hran jsou zapsány čísla u hran. Šipky udávají směr hran, tedy směr proudění uvažované substance po spojnicích. (Pokud směr proudění není důležitý, vedeme mezi vrcholy dvojici opačně orientovaných hran se stejnou kapacitou.) Kapacity hran pak omezují maximální množství přenesené substance.

**Poznámka:** V praxi může být zdrojů a stoků více, ale v definici stačí pouze jeden zdroj a stok, z něhož / do něž vedou hrany do ostatních zdrojů / stoků. (Dokonce pak různé zdroje a stoky mohou mít své kapacity.)

Obvykle nás na síti nejvíce zajímá, kolik nejvíce substance můžeme (různými cestami) přenést ze zdroje do stoku. Pro to musíme definovat pojem toku, což je formální popis okamžitého stavu přenášení v síti.

**Značení:** Pro jednoduchost píšeme ve výrazech znak  $e \rightarrow v$  pro hranu  $e$  přicházející do vrcholu  $v$  a  $e \leftarrow v$  pro hranu  $e$  vycházející z  $v$ .

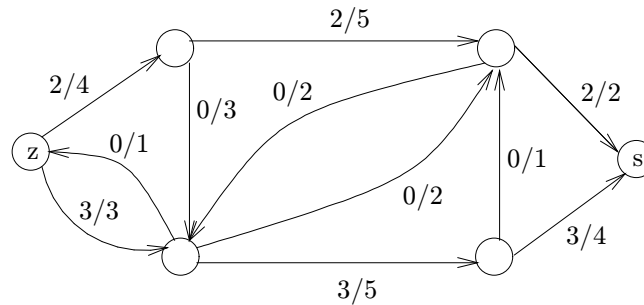
**Definice 11.2.** **Tok** v síti  $S = (G, z, s, w)$  je funkce  $f : E(G) \rightarrow \mathbb{R}_0^+$  splňující

- $\forall e \in E(G) : 0 \leq f(e) \leq w(e)$ ,
- $\forall v \in V(G), v \neq z, s : \sum_{e \rightarrow v} f(e) = \sum_{e \leftarrow v} f(e)$ .

**Velikost toku**  $f$  je dána výrazem  $\|f\| = \sum_{e \leftarrow z} f(e) - \sum_{e \rightarrow z} f(e)$ .

**Značení:** Tok a kapacitu hran v obrázku sítě budeme zjednodušeně zapisovat ve formátu  $F/C$ , kde  $F$  je hodnota toku na hraně a  $C$  je její kapacita.

**Komentář:** Neformálně tok znamená, kolik substance je každou hranou zrovna přenášeno (ve směru této hrany, proto hrany musí být orientované). Tok je pochopitelně nezáporný a dosahuje nejvýše dané kapacity hrany.



Ve vyobrazeném příkladě vede ze zdroje vlevo do stoku vpravo tok o celkové velikosti 5.

**Poznámka:** Obdobně se dá velikost toku definovat u stoku, neboť

$$0 = \sum_e (f(e) - f(e)) = \sum_v \sum_{e \leftarrow v} f(e) - \sum_v \sum_{e \rightarrow v} f(e) = \sum_{v=z,s} \left( \sum_{e \leftarrow v} f(e) - \sum_{e \rightarrow v} f(e) \right).$$

(Dvojitě sumy uprostřed předchozího vztahu nabývají stejných hodnot pro všechny vrcholy kromě  $z$  a  $s$  dle definice toku.) Proto velikost toku počítaná u zdroje je rovna opačné velikosti toku počítaného u stoku

$$\left( \sum_{e \leftarrow z} f(e) - \sum_{e \rightarrow z} f(e) \right) = - \left( \sum_{e \leftarrow s} f(e) - \sum_{e \rightarrow s} f(e) \right).$$

## 11.2 Hledání maximálního toku

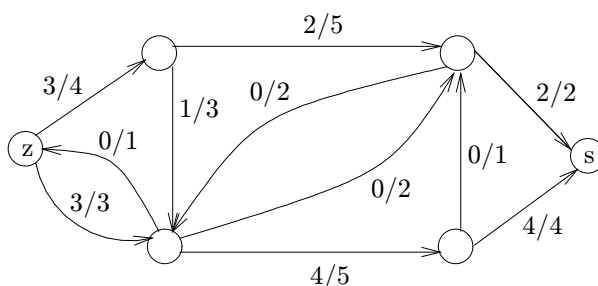
Naším úkolem je najít co největší tok v dané síti. Pro jeho nalezení existují jednoduché a velmi rychlé algoritmy.

### Problém 11.3. O maximálním toku v síti

Je dána síť  $S = (G, z, s, w)$  a našim úkolem je pro ni najít co největší tok ze zdroje  $z$  do stoku  $s$  vzhledem k ohodnocení  $w$ .

Formálně hledáme  $\max \|f\|$  dle Definice 11.2.

**Komentář:** Tok velikosti 5 uvedený v ukázce v předchozí části nebyl optimální, neboť v té síti najdeme i tok velikosti 6:



Jak však poznáme, že větší tok již v dané síti neexistuje? V této konkrétní ukázce to není obtížné, vidíme totiž, že obě dvě hrany přicházející do stoku mají součet kapacit  $2 + 4 + 6$ , takže více než 6 do stoku ani přitéct nemůže. V obecnosti lze použít obdobnou úvahu, kdy najdeme podmnožinu hran, které nelze tokem “obejít” a které v součtu kapacit dají velikost našeho toku. Existuje však taková množina hran vždy? Odpověď nám dá následující definice a věta.

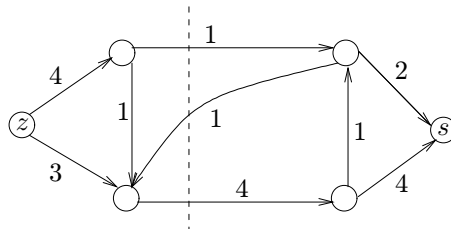
### Definice 11.4. Řez v síti $S = (G, z, s, w)$

je podmnožina hran  $C \subseteq E(G)$  taková, že v podgrafu  $G - C$  (tj. po odebrání hran  $C$  z  $G$ ) nezůstane žádná orientovaná cesta ze  $z$  do  $s$ .

*Velikost* řezu  $C$  rozumíme součet kapacit hran z  $C$ , tj.  $\|C\| = \sum_{e \in C} w(e)$ .

### Věta 11.5. Maximální velikost toku v síti je rovna minimální velikosti řezu.

**Komentář:** Na následujícím obrázku vidíme trochu jinou síť s ukázkou netriviálního minimálního řezu velikosti 5, naznačeného svislou čárkovanou čarou. Všimněte si dobře, že definice řezu mluví o přerušení všech orientovaných cest ze  $z$  do  $s$ , takže do řezu stačí započítat hrany jdoucí přes svislou čáru od  $z$  do  $s$ , ale ne hranu jdoucí zpět. Proto je velikost vyznačeného řezu  $1 + 4 = 5$ .



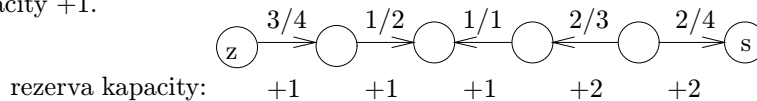
**Poznámka:** Tato věta poskytuje tzv. *dobrou charakterizaci* problému maximálního toku: Když už nalezneme maximální tok, tak je pro nás vždy snadné dokázat, že lepší tok není, nalezením příslušného řezu o stejné velikosti. Přitom toto zdůvodnění řezem můžeme směle ukázat i někomu, kdo se vůbec nevyzná v matematice.

Důkaz Věty 11.5 bude proveden následujícím algoritmem.

**Definice:** Mějme síť  $S$  a v ní tok  $f$ . *Nenasycená cesta* (v  $S$  vzhledem k  $f$ ) je neorientovaná cesta v  $G$  z vrcholu  $u$  do vrcholu  $v$  (obvykle ze  $z$  do  $s$ ), tj. posloupnost navazujících hran  $e_1, e_2, \dots, e_m$ , kde  $f(e_i) < w(e_i)$  pro  $e_i$  ve směru z  $u$  do  $v$  a  $f(e_i) > 0$  pro  $e_i$  v opačném směru.

Hodnotě  $w(e_i) - f(e_i)$  pro hrany  $e_i$  ve směru z  $u$  do  $v$  a hodnotě  $f(e_i)$  pro hrany  $e_i$  v opačném směru říkáme *rezerva kapacity* hran  $e_i$ . Nenasycená cesta je tudíž cesta s kladnými rezervami kapacit všech hran.

**Komentář:** Zde vidíme příklad nenasyčené cesty ze zdroje do stoku s minimální rezervou kapacity +1.



Všimněte si dobře, že cesta není orientovaná, takže hrany na ní jsou v obou směrech.

**Algoritmus 11.6. Ford–Fulkersonův pro tok v síti**

**vstup** síť  $S = (G, z, s, w)$ ;

**tok**  $f \equiv 0$ ;

**do** {

*prohledáváním grafu najdeme množinu  $U$  vrcholů  $G$ ,  
    do kterých se dostaneme ze  $z$  po nenasyčených cestách;*

**if** ( $s \in U$ ) {

*$P =$  (výše nalezená) nenasyčená cesta v  $S$  ze  $z$  do  $s$ ;  
        zvětšíme tok  $f$  o minimální rezervu kapacity hran v  $P$ ;*

    }

**while** ( $s \in U$ );

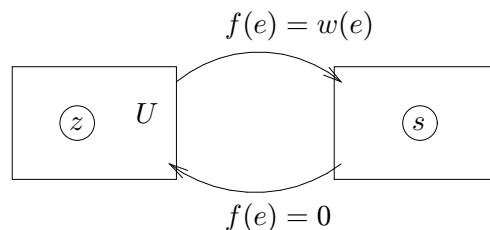
**výstup** vypíšeme maximální tok  $f$ ;

**výstup** vypíšeme min. řez jako množinu hran vedoucích z  $U$  do  $V(G) - U$ .

**Důkaz správnosti Algoritmu 11.6:**

Pro každý tok  $f$  a každý řez  $C$  v síti  $S$  platí  $\|f\| \leq \|C\|$ . Jestliže po zastavení algoritmu s tokem  $f$  nalezneme v síti  $S$  řez o stejné velikosti  $\|C\| = \|f\|$ , je jasné, že jsme našli maximální možný tok v síti  $S$ . Zároveň tím dokážeme i platnost Věty 11.5.

Takže stačí dokázat, že po zastavení algoritmu nastane rovnost  $\|f\| = \|C\|$ , kde  $C$  je vypsaný řez mezi  $U$  a zbytkem grafu  $G$ . Vezměme tok  $f$  v  $S$  bez nenasyčené cesty ze  $z$  do  $s$ . Pak množina  $U$  z algoritmu neobsahuje  $s$ . Schematicky vypadá situace takto:



Jelikož z  $U$  žádné nenasyčené cesty dále nevedou, má každá hrana  $e \leftarrow U$  (odcházející z  $U$ ) plný tok  $f(e) = w(e)$  a každá hrana  $e \rightarrow U$  (přicházející do  $U$ ) tok  $f(e) = 0$ . Velikost toku  $f$  ze  $z$  do  $s$  se také dá psát jako

$$\|f\| = \sum_{e \leftarrow U} f(e) - \sum_{e \rightarrow U} f(e) = \sum_{e \leftarrow U} f(e) = \sum_{e \in C} w(e) = \|C\|.$$

To je přesně, co jsme chtěli dokázat o výsledném toku. □

Z popisu Algoritmu 11.6 vyplývá ještě jeden důležitý důsledek:

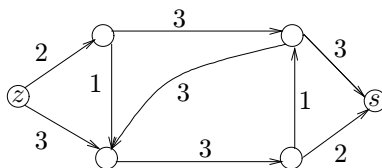
**Důsledek 11.7.** Pokud jsou kapacity hran sítě  $S$  celočíselné, optimální tok také vyjde celočíselně.

**Poznámka:** Algoritmus pro celá čísla kapacit vždy skončí. Pro reálná čísla se ale dají najít extrémní případy, které nepovedou k řešení ani v limitě.

Pro rychlý běh algoritmu je vhodné hledat nejkratší nenasycenou cestu, tj. prohledáváním sítě do šířky. V takové implementaci algoritmus dobře a rychle funguje i s reálnými kapacitami hran.

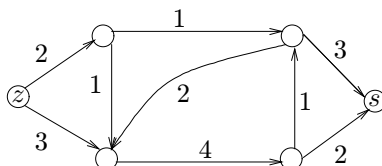
### Úlohy k řešení

(11.2.1) Jaký je maximální tok touto sítí ze  $z$  do  $s$ ?



(11.2.2) Co obsahuje výsledná množina  $U$  Algoritmu 11.6 v předchozím příkladě?

(11.2.3) Kde je minimální řez v této sítí mezi  $z$  a  $s$ ?

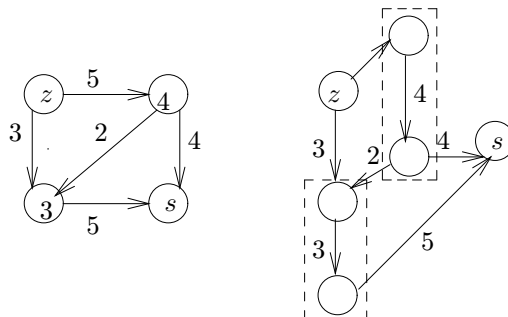


### 11.3 Zobecnění sítě a další aplikace

Pojmy sítě a toků v ní lze zobecnit v několika směrech. My si zde stručně uvedeme tři možnosti:

1. U sítě můžeme zadat i *kapacity vrcholů*.

To znamená, že žádným vrcholem nemůže celkem protéct více než povolené množství substance. Takovou síť “zdvojením” vrcholů snadno převedeme na běžnou síť, ve které kapacity původních vrcholů budou uvedeny u nových hran spojujících zdvojené vrcholy. Viz neformální schéma:



2. Pro hrany sítě lze zadat také *minimální kapacity*, tedy dolní meze toku.

(Například u potrubní sítě mohou minimální vyžadované průtoky vody garantovat, že nedojde k zanesení potrubí.) V této modifikaci úlohy již přípustný tok nemusí vůbec existovat. Takto zobecněná úloha je také snadno řešitelná, ale my se jí nebudeme zabývat.

3. V síti lze najednou přepravovat více substancí. To vede na problém tzv. *vícekomoditních toků* v síti. Tento problém je složitější a už není v obecnosti snadno řešitelný, a proto se jím nebudeme zabývat.

Kromě uvedených (a podobných) zobecnění toků v sítích jsou velmi zajímavé i některé speciální formulace problému toků, které se vyskytují v možná i nečekaných oblastech. Více o tom napíšeme v dalších částech tohoto oddílu.

### Bipartitní párování

V Lekci 10 jsme definovali bipartitní grafy jako ty, které lze korektně obarvit dvěma barvami. Jinými slovy to jsou grafy, jejichž vrcholy lze rozdělit do dvou množin tak, že všechny hrany vedou mezi těmito množinami.

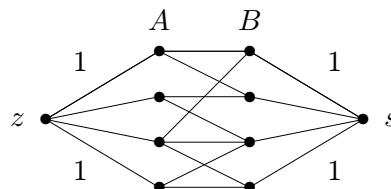
**Definice:** *Párování* v (bipartitním) grafu  $G$  je podmnožina hran  $M \subseteq E(G)$  taková, že žádné dvě hrany z  $M$  nesdílejí koncový vrchol.

**Komentář:** Pojem (bipartitního) párování má přirozenou motivaci v mezilidských vztazích. Pokud neuvažujeme bigamii ani jisté menšiny, můžeme si partnerské vztahy představit jako párování v bipartitním grafu. Jednu stranu grafu tvoří muži a druhou ženy. Hrana mezi mužem a ženou znamená vzájemné sympatie (přitom jedinec může mít vzájemné sympatie s několika jinými opačného pohlaví). Pak skutečné partnerské vztahy představují párování v popsaném grafu.

Úlohu nalézt v daném bipartitním grafu co největší párování lze poměrně snadno vyřešit pomocí toků ve vhodně definované síti. Uvedená metoda použití toků v síti na řešení problému párování přitom hezky ilustruje obecný přístup, jakým toky v sítích pomohou řešit i úlohy, které na první pohled se sítěmi nemají nic společného.

#### Metoda 11.8. *Nalezení bipartitního párování*

Pro daný bipartitní graf  $G$  s vrcholy rozdělenými do množin  $A, B$  sestrojíme síť  $S$  následovně:



Všechny hrany sítě  $S$  orientujeme od zdroje do stoku a přiřadíme jim kapacity 1. Nyní najdeme (celočíslný) maximální tok v  $S$  Algoritmem 11.6. Do párování vložíme ty hrany grafu  $G$ , které mají nenulový tok.

**Důkaz správnosti Metody 11.8:** Podle Důsledku 11.7 bude maximální tok celočíselný, a proto každou hranou poteče buď 0 nebo 1. Jelikož však do každého vrcholu v  $A$  může ze zdroje přitéct jen tok 1, bude z každého vrcholu  $A$  vybrána do párování nejvýše jedna hrana. Stejně tak odvodíme, že z každého vrcholu  $B$  bude vybrána nejvýše jedna hrana, a proto vybraná množina skutečně bude párováním. Zároveň to bude největší možné párování, protože z každého párování lze naopak vytvořit tok příslušné velikosti a větší než nalezený tok v  $S$  neexistuje.  $\square$

**Poznámka:** Popsaná metoda je základem tzv. Maďarského algoritmu pro párování v bipartitních grafech. Úlohu nalezení maximálního párování lze definovat i pro obecné grafy a také ji efektivně algoritmicky vyřešit, ale příslušný algoritmus není jednoduchý.

## Vyšší grafová souvislost

Představme si, že na libovolném grafu  $G$  definujeme zobecněnou síť tak, že kapacity všech hran a všech vrcholů položíme rovny 1 v obou směrech. Pak celočíselný tok (viz Důsledek 11.7) velikosti  $k$  mezi dvěma vrcholy  $u, v$  se skládá ze soustavy  $k$  disjunktních cest (mimo společné koncové vrcholy  $u, v$ ). Naopak řez odděluje  $u$  a  $v$  do různých souvislých komponent zbylého grafu. Aplikace Věty 11.5 na tuto situaci přímo poskytne následující tvrzení.

**Lema 11.9.** *Nechť  $u, v$  jsou dva vrcholy grafu  $G$  a  $k > 0$  je přirozené číslo. Pak mezi vrcholy  $u$  a  $v$  existuje v  $G$  aspoň  $k$  disjunktních cest, právě když po odebrání libovolných  $k-1$  vrcholů různých od  $u, v$  z  $G$  zůstanou  $u$  a  $v$  ve stejné komponentě souvislosti zbylého grafu.*

Použitím tohoto tvrzení pro všechny dvojice vrcholů grafu snadno dokážeme dříve uvedenou důležitou Větu 7.5 (Mengerovu). Ještě jiné použití si ukážeme na problému výběru reprezentantů množin.

**Definice:** Nechť  $M_1, M_2, \dots, M_k$  jsou neprázdné množiny. *Systémem různých reprezentantů* množin  $M_1, M_2, \dots, M_k$  nazýváme takovou posloupnost různých prvků  $(x_1, x_2, \dots, x_k)$ , že  $x_i \in M_i$  pro  $i = 1, 2, \dots, k$ .

Důležitým a dobře známým výsledkem v této oblasti je Hallova věta plně popisující, kdy lze systém různých reprezentantů daných množin nalézt.

**Věta 11.10.** *Nechť  $M_1, M_2, \dots, M_k$  jsou neprázdné množiny. Pro tyto množiny existuje systém různých reprezentantů, právě když platí*

$$\forall J \subseteq \{1, 2, \dots, k\} : \left| \bigcup_{j \in J} M_j \right| \geq |J|,$$

*neboli pokud sjednocení libovolné skupiny z těchto množin má alespoň tolik prvků, kolik množin je sjednoceno.*

**Důkaz:** Označme  $x_1, x_2, \dots, x_m$  po řadě všechny prvky ve sjednocení  $M_1 \cup M_2 \cup \dots \cup M_k$ . Definujeme si bipartitní graf  $G$  na množině vrcholů  $\{1, 2, \dots, k\} \cup \{x_1, x_2, \dots, x_m\} \cup \{u, v\}$ , ve kterém jsou hrany  $\{u, i\}$  pro  $i = 1, 2, \dots, k$ , hrany  $\{v, x_j\}$  pro  $j = 1, 2, \dots, m$  a hrany  $\{i, x_j\}$  pro všechny dvojice  $i, j$ , pro které  $x_j \in M_i$ .

**Komentář:** Konstrukce našeho grafu  $G$  je obdobná konstrukci sítě v Metodě 11.8: Vrcholy  $u$  a  $v$  odpovídají zdroji a stoku, ostatní hrany přicházející do vrcholu  $x_j$  znázorňují všechny z daných množin, které obsahují prvek  $x_j$ .

Cesta mezi  $u$  a  $v$  má tvar  $u, i, x_j, v$ , a tudíž ukazuje na reprezentanta  $x_j \in M_i$ . Systém různých reprezentantů tak odpovídá  $k$  disjunktním cestám mezi  $u$  a  $v$ . Nechť  $X$  je nyní libovolná minimální množina vrcholů v  $G$ , neobsahující samotné  $u$  a  $v$ , po jejímž odebrání z grafu nezbude žádná cesta mezi  $u$  a  $v$ . Podle Lematu 11.9 a této úvahy mají naše množiny systém různých reprezentantů, právě když každá taková oddělující množina  $X$  má aspoň  $k$  prvků.

Položme  $J = \{1, 2, \dots, k\} \setminus X$ . Pak každá hrana z  $J$  (mimo  $u$ ) vede do vrcholů z  $X \cap \{x_1, \dots, x_m\}$  (aby nevznikla cesta mezi  $u, v$ ), a proto

$$\left| \bigcup_{j \in J} M_j \right| = |X \cap \{x_1, \dots, x_m\}| = |X| - |X \cap \{1, \dots, k\}| = |X| - k + |J|.$$

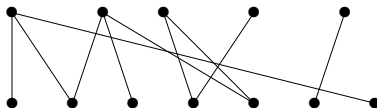
Vidíme tedy, že  $|X| \geq k$  pro všechny (minimální) volby oddělující  $X$ , právě když  $\left| \bigcup_{j \in J} M_j \right| \geq |J|$  pro všechny volby  $J$ , což je dokazovaná podmínka naší věty.  $\square$



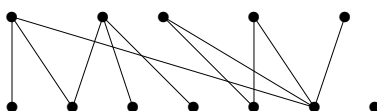
Poznámka: Předchozí důkaz nám také dává návod, jak systém různých reprezentantů pro dané množiny nalézt – stačí použít Algoritmus 11.6 na vhodně odvozenou síť.

### Úlohy k řešení

(11.3.1) Najděte největší párování v tomto bipartitním grafu:



(11.3.2) Najděte největší párování v tomto bipartitním grafu:



(11.3.3) Mají všechny tříprvkové podmnožiny množiny  $\{1, 2, 3, 4\}$  systém různých reprezentantů?

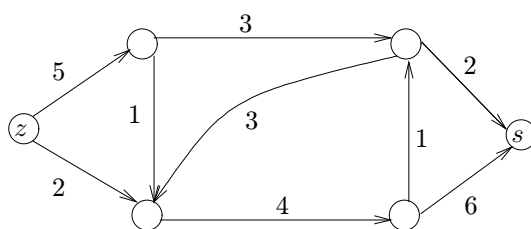
(11.3.4) Mají všechny tříprvkové podmnožiny množiny  $\{1, 2, 3, 4, 5\}$  systém různých reprezentantů?

### Rozšiřující studium

Problematika toků v sítích je běžnou součástí teorie grafů (i když není pokryta v [5]) a je možno najít její obsáhlejší popis třeba v [3] nebo v [7].

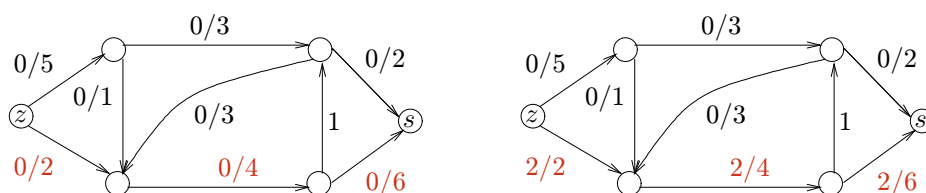
## 11.4 Cvičení: Příklady toků v sítích

**Příklad 11.11.** Jaký je maximální tok touto sítí ze  $z$  do  $s$ ?

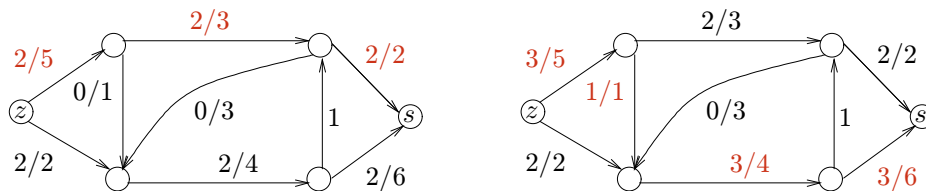


Na tomto příkladě si ukážeme průběh Algoritmu 11.6.

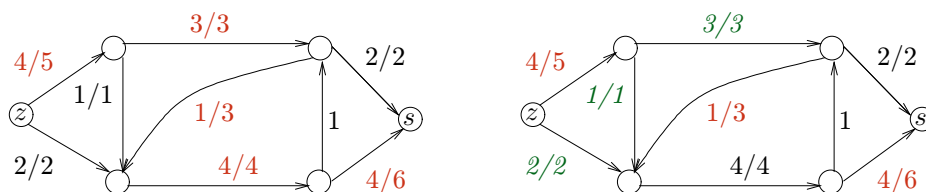
Začneme s nulovým tokem a najdeme nejkratší z nenasycených cest mezi  $z$  a  $s$  (vedoucí spodem grafu a vyznačenou na prvním obrázku). Minimální rezerva kapacity na této cestě je 2, takže tok nasatíme o 2 podél této cesty (viz obrázek vpravo).



V dalším kroku najdeme nenasyčenou cestu délky 3 vedoucí vrchem grafu. Její minimální rezerva kapacity je opět 2, takže ji o tolik nasytíme (viz další obrázek vlevo). Nyní již nenasyčená cesta délky 3 neexistuje, takže najdeme nenasyčenou cestu délky 4 s rezervou kapacity 1, kterou hned nasytíme (viz obrázek vpravo).



Obdobně ještě najdeme nenasyčenou cestu délky 5, kterou také nasytíme o 1. Závěrečný obrázek nám ukazuje všechny nenasyčené hrany, mezi kterými již nevede žádná nenasyčená cesta ze  $z$  do  $s$ , takže máme maximální tok velikosti 6 v naší síti.

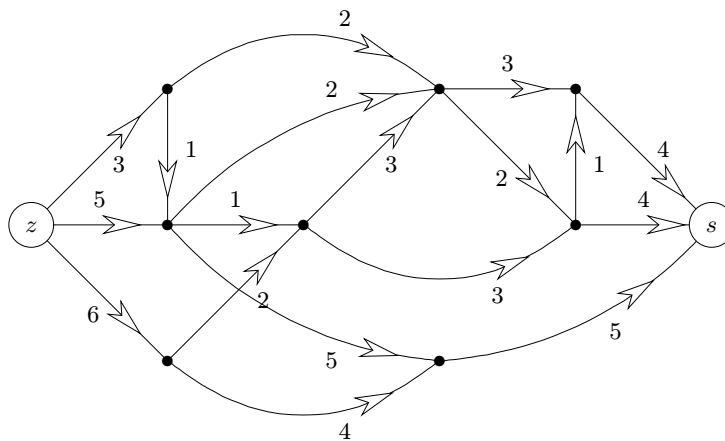


Zároveň je na posledním obrázku vyznačen i příslušný minimální řez velikosti 6. □

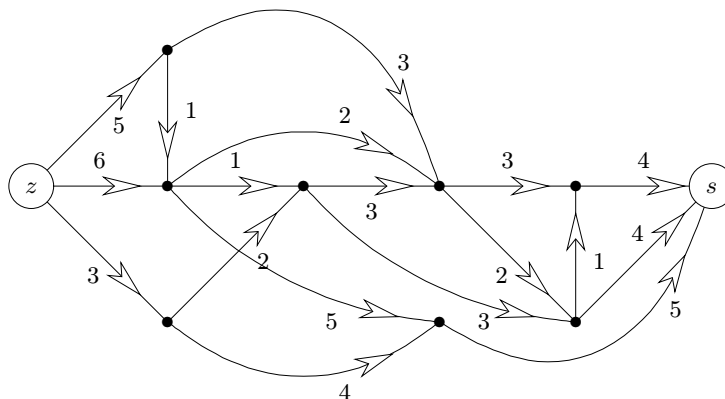
### Úlohy k řešení

(11.4.1) Najděte maximální tok v následující síti. (Kapacity hran jsou vyznačeny u svých šipek.)

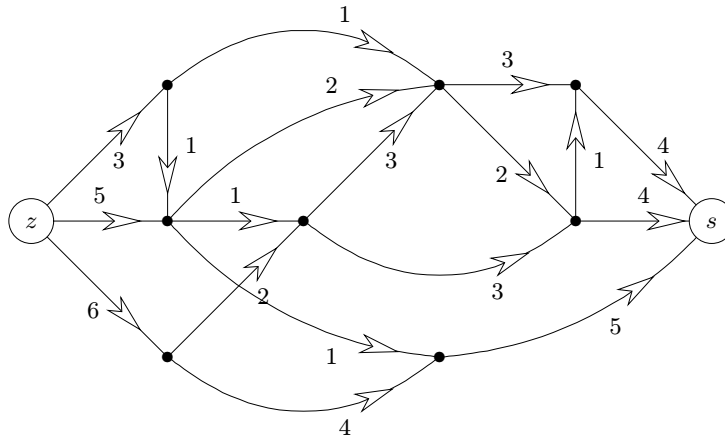
Tok do obrázku zapište, vyznačte hrany příslušného minimálního řezu a napište velikost toku.



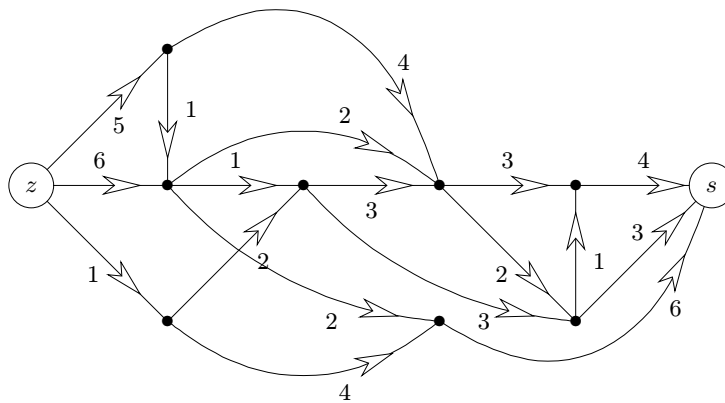
(11.4.2) Najděte maximální tok v následující síti.



(11.4.3) Najděte maximální tok v následující síti.



(11.4.4) Najděte maximální tok v následující síti.



(11.4.5) Má tento seznam množin systém různých reprezentantů? Pokud ano, vyznačte je v množinách, jinak správně zdůvodněte, proč systém různých reprezentantů nemají.

- |            |            |          |
|------------|------------|----------|
| {1,2,3,4}, | {6,7,8,9}, | {3,5,7}, |
| {3,4,5,6}, | {1,2,8,9}, | {3,6,7}, |
| {4,5,6,7}, | {4,5,6},   | {4,6,7}. |

(11.4.6) Má tento seznam množin systém různých reprezentantů? Pokud ano, vyznačte je v množinách, jinak správně zdůvodněte, proč systém různých reprezentantů nemají.

- |            |            |          |
|------------|------------|----------|
| {1,2,3,4}, | {6,7,8,9}, | {1,3,9}, |
| {3,4,5,6}, | {1,2,3,9}, | {2,4,9}, |
| {4,5,6,7}, | {2,3,4},   | {3,4,9}. |

## Část III

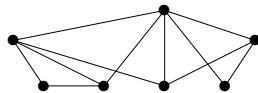
### Klíč k řešení úloh

- (1.1.1) 91  
(1.1.2) 990  
(1.1.3) 3  
(1.1.4)  $-4$   
(1.1.5) 4  
(1.1.6)  $-3$   
(1.1.7) 16  
(1.1.8) 12  
(1.1.9)\*  $\lfloor x + 0.5 \rfloor$   
(1.2.1)  $n(6n - 1)(6n - 2)$   
(1.2.2) 720, permutace  
(1.2.3) 210, kombinace  
(1.2.4) 420, variace  
(1.2.5) Nezávislé dva výběry, takže  $49 \cdot 49$ .  
(1.2.6)\* Nyní místo původních dvou dvojic  $x, y$  a  $y, x$  máme jen jednu bez pořadí. Výsledek však není prostě polovinou  $49 \cdot 49/2$ , neboť dvojici stejných čísel  $x, x$  dělit nemůžeme. Správně je  $C(49, 2) + 49 = 1225$  za neuspořádané dvojice různých čísel plus 49 dvojic stejných čísel.  
(1.2.7)\* Nezávislými výběry kombinací  $\binom{13}{3} \cdot \binom{9}{2}$ .  
(1.3.1) Permutace s opakováním  $3 \times A, 2 \times N$ , výsledek 3360.  
(1.3.2)  $\frac{12!}{3!^4}$   
(1.3.3) Kombinace s opakováním  $C^*(3, 7) = 36$ .  
(1.3.4)\* Kombinace s opakováním  $C^*(4, 6) = 84$ .  
(1.3.5)\* To už je jiná úloha než předchozí, neboť i ve výsledku stále rozeznáme první, druhou, atd. Pro první kuličku máme 4 možnosti, pro druhou nezávisle zase 4, pro třetí také ..., celkem  $4^6 = 4096$ .  
(1.4.1) Možné součty jsou 3, 4, ..., 28, tedy 26 možností.  
(1.4.2) 10  
(1.4.3)  $\binom{9}{4} \cdot 3$  – nejprve vybereme čtyři, co budou hrát, a pak je rozdělíme třemi způsoby na dvě dvojice proti sobě.  
(1.4.4) Nezávislé výběry pro uspořádání čtveřice králů, pak čtveřice dam, atd., celkem  $4!^8$ .  
(1.4.5)\*  $C^*(4 + 1, 10)$ , k 4 barvám je nutno přičíst 1 za zbylé bílé kuličky.  
(1.4.6)\*  $C(\ell - n, n)$   
(2.2.1)  $\frac{1}{2}$   
(2.2.2)  $\frac{4}{36} = \frac{1}{9}$   
(2.2.3)  $\frac{4}{32} = \frac{1}{8}$   
(2.3.1) Nejsou, například pokud jeden dostane pikové eso, druhý ho už dostat nemůže.  
(2.3.2) Ano, je. Ať na první padne cokoliv, druhá má stejnou šanci se trefit do stejného čísla.  
(2.3.3)  $3! \cdot \frac{1}{6^3} = \frac{1}{36}$   
(2.3.4) Tři nezávislé hody,  $(\frac{1}{2})^5 = \frac{1}{32}$ .  
(2.3.5) Královský pokr může dostat 28 způsoby z  $\binom{32}{5}$  (neuspořádaných) rozdání, takže  $28/\binom{32}{5}$  pro jedno rozdání a  $[28/\binom{32}{5}]^2$  pro obě.  
(2.4.1)  $\frac{10}{6}$   
(2.4.2) 7, pro součet nepotřebujeme nezávislost. Navíc kostky obvykle mají součet protějších stěn po 7.

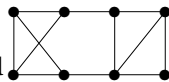
- (2.4.3)  $\frac{1}{2} \cdot 2 + \frac{1}{2} \cdot 3 = 2.5$
- (2.4.4)\*  $\frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 3 + \frac{1}{16} \cdot 4 + \dots = 2$
- (2.5.1)  $\frac{1}{120}$ , kombinace!
- (2.5.2)  $\frac{1}{24}$
- (2.5.3)  $\frac{5}{2^5}$
- (2.5.4)  $\frac{1}{8}$
- (2.6.1)  $\frac{1}{10}$
- (2.6.2)  $\frac{4 \cdot 3}{32 \cdot 31} = \frac{3}{248}$
- (2.6.3)  $\frac{4!}{32 \cdot 31 \cdot 30 \cdot 29}$
- (2.6.4)  $\binom{6}{3}/2^6$  – vybereme všechny posloupnosti výsledků se 3 hlavami a 3 orly, každá z nich má pravděpodobnost  $1/2^6$ .
- (2.6.5) Výsledky: a)  $\frac{4 \cdot 3}{32 \cdot 31}$ , b)  $\frac{4 \cdot 4}{32 \cdot 31}$ , c)  $\frac{28 \cdot 27}{32 \cdot 31}$ , d)  $\frac{32 \cdot 3}{32 \cdot 31}$
- (2.6.6) a)  $((\binom{17}{2} + \binom{7}{2})/\binom{24}{2})$ , b)  $(17 \cdot 7)/\binom{24}{2}$ .
- (2.6.7)  $\binom{28}{5}/\binom{32}{5} = \frac{1755}{3596}$
- (2.6.8)  $\frac{1}{6} \cdot \frac{1}{3} + \frac{1}{3} \cdot \frac{1}{6} + \frac{1}{6} \cdot \frac{1}{6} = \frac{5}{36}$
- (2.6.9)\*  $\frac{21}{6} + \frac{21}{36} + \frac{21}{216} = \frac{301}{72} = 4.18$
- (2.6.10)\*  $E = 2 \cdot \frac{5}{17} + 3 \cdot (1 - \frac{5}{17} - \frac{12}{17} \cdot \frac{6}{16}) + 4 \cdot \frac{12}{17} \cdot \frac{6}{16}$
- (2.6.11)\* Pokud na první kostce padne 1, nenastane součet  $s \geq 8$ , naopak pokud padne 6, nenastane součet  $s \leq 6$ . Proto jedině jev součtu  $B_7$  může být nezávislý na jevu  $A_k$ . Dosazením do definice nezávislosti ověříme, že skutečně  $A_k$  a  $B_7$  jsou nezávislé pro všechna  $k = 1, \dots, 6$ .
- (2.6.12)\*  $((\binom{10}{5})/2^{10})^4$ , stačí, že v každé barvě budou póly přesně napůl.
- (3.5.1)  $\binom{2n+1}{n-1}$
- (3.5.2)\*  $\binom{n+4}{3}$  (Uvědomte si nejprve, že  $\binom{n}{0} = \binom{n+1}{0}$ .)
- (3.6.1) Ano, posledních čtyřčísí je 10000 a studentů kolem 17000.
- (3.6.2) Ano,  $\binom{12}{4} < 500$ .
- (3.7.1) Indukcí, nebo rozkladem  $n^3 - n = (n-1)n(n+1)$ .
- (3.7.2) Pro  $n \geq 6$ , důkaz třeba indukcí podle  $n$ , nebo úpravou.
- (3.7.3) Indukcí podle  $n$ .
- (3.7.4)\* Rozepsáním faktoriálu na součin a úpravou členů.
- (3.7.5)\*  $(n-1)3^{n+1} + 3 + (2(n+1)-1)3^{n+1} = (n-1)3^{n+1} + (2n+1)3^{n+1} + 3 = n3^{n+2} + 3$
- (4.1.1) 5
- (4.1.2) 10
- (4.1.3) 14
- (4.1.4)\* Reflexivní a symetrická, ale není tranzitivní.
- (4.2.1) a) délky 4 b) nejmenší 1, největší není c) 6
- (4.2.2) 8 tříd (s minimy 1, ..., 8), nejvíce je s minimem 1 – celkem  $\binom{11}{4}$ .
- (4.4.1) (1, 4, 6, 2, 5)(3)
- (4.4.2) 6-krát.
- (4.4.3)\*  $(n-1)!$
- (4.5.1) 25
- (4.5.2) 17
- (4.6.1) Reflexivní, antisymetrická, tranzitivní, tedy částečné uspořádání. Lineární není, protože dvě slova nemusí být porovnatelná.
- (4.6.2) Reflexivní, symetrická, tranzitivní, tedy ekvivalence.
- (4.6.3) a) je to ekvivalence, b) je reflexivní, symetrická, ale ne tranzitivní, c) je reflexivní, symetrická, ale ne tranzitivní, d) je to ekvivalence
- (4.6.4)\* a) je to částečné uspořádání, b) není symetrická, antisymetrická ani tranzitivní, c) je tranzitivní, ale ne symetrická ani antisymetrická, d) je to částečné uspořádání
- (4.6.5)\* Je to ekvivalence – přesně podle zbytkových tříd modulo 3.

- (4.6.6) čtyř:  $(1, 7)(2)(3)(4, 6, 5)$   
 (4.6.7)  $(1)(2)(3)(4, 5, 6, 7)$   
 (4.6.8)  $(1)(2, 3)(4, 7)(5)(6)$   
 (4.6.9)  $(1, 6, 8, 2, 4)(3, 5, 7)$   
 (4.6.10) Až 9, když složením vznikne identická permutace.  
 (4.6.11) 15  
 (4.6.12) 8  
 (4.6.13)\* 5, musíte ručně pokrátit jmenovatele zlomku.

- (6.1.1) Kružnice délky 5.  
 (6.1.2) Pro  $n = 1$  a 2, cesty délky 0 a 1.  
 (6.1.3) Pro  $n = 3$ , trojúhelník.  
 (6.1.4) Pro  $m = n = 2$ , délky čtyři.  
 (6.1.5) Pro  $m = 1$  nebo  $n = 1$  a druhé libovolné.  
 (6.1.6) 9  
 (6.1.7) Oba stejně 36 hran.  
 (6.2.1) 34



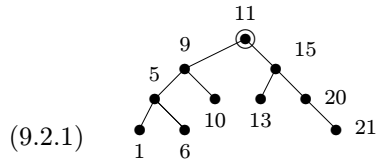
- (6.2.2) Ano:  
 (6.2.3) Ano, kružnice délky 6 a dvě kružnice délky 3 vedle sebe.  
 (6.3.1) 3, více nelze, neboť na obvodě kružnice by se musely ob-jeden střídát.  
 (6.3.2) 3  
 (6.3.3) 7  
 (6.3.4)\* 5, dokážete ji vůbec najít?



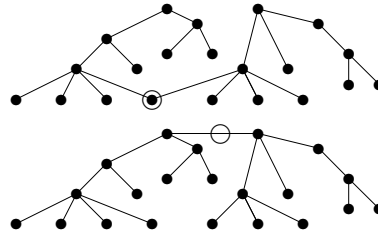
- (6.3.5) Ano, například . Tento graf obsahuje 4 trojúhelníky, kdežto původní graf jen 3. Mohli jste však sestrojít úplně jiný graf...  
 (6.3.6) Jeden, jen kružnice délky 5.  
 (6.6.1) Ano.  
 (6.6.2) Ne.  
 (6.6.3) Dva.  
 (6.6.4) A: 23, B: 20, C: 22.  
 (6.6.5) Snadno...  
 (6.6.6) Isom. z pravého do levého grafu: spodní dva nahoru vpravo, pravé dva dolů vpravo, atd.  
 (6.6.7) ...  
 (6.6.8)\* A ano,  $K_{3,3}$  a graf trojbokého hranolu. B ne, protože dva vrcholy stupně 3 musí být spojené s ostatními a tím již získáme celý graf jednoznačně.  
 (6.6.9)\*  $B \simeq C$ ,  $A \simeq D$ , u neisomorfních dvojic vždy jedna obsahuje kružnice délky 5 a druhá ne.  
 (6.6.10) Nejdelší  $C_8$  v obou, nejdelší indukovaná  $C_6$  v  $A$  a jen  $C_5$  v  $B$ .  
 (6.6.11)\*  $\binom{10}{4} \cdot 3$ , neboť  $\binom{10}{4}$  způsoby vybereme čtveřici vrcholů pro ten podgraf a každé 4 vrcholy lze třemi způsoby propojit do kružnice.  
 (6.6.12)\* Je jich 10, 6 z nich vznikne různým přidáváním vrcholů stupně 2 na hrany úplného grafu  $K_4$  a 4 další jsou jiné.  
 (6.6.13) ...  
 (6.6.14) Vlevo 4, vpravo 3.

- (7.1.1) B  
 (7.1.2) 4  
 (7.3.1) Vrcholově  $n$ -souvislý.

- (7.3.2) Najdeme 3 disjunktní cesty mezi  $x, y$ . Proto musíme vypustit 3 vrcholy.
- (7.3.3) 1 do kružnice.
- (7.4.1) Ne, má čtyři vrcholy lichého stupně.
- (7.4.2) Tři, dvě by teoreticky stačily na změnu všech stupňů na sudé, ale v tomto grafu jen dvě hrany prostě přidat nejdu.
- (7.5.1) Jednu.
- (7.5.2) 5, samé trojúhelníky.
- (7.5.3)\* 6, samé úplné grafy  $K_5$ .
- (7.5.4) Celkem 4.
- (7.5.5) Souvislý s 15 hranami (ke každému vrcholu 3 sousedé).
- (7.5.6) 18, stupně  $\geq 3$ , třeba jako šestiboký hranol.
- (7.5.7) a) 28:  $K_8 + K_1 + K_1$ , b) 9:  $K_3 + K_3 + K_3 + K_1$
- (7.5.8)\* Jen v A: graf  $K_{3,3}$  a graf  $C_5$  s jednou diagonálou. Pro B uvažte doplněk toho grafu – má stupně 2, 1, 1, 1, 1, což dá jednoznačný graf.
- (7.5.9)\* Jen v A: cesta délky 4 a trojúhelník s hranou vedle.
- (7.5.10)\* Jen B: úplný  $K_4$  s hranou vedle a dále  $K_4$  s “rozpojenou” hranou. Pro A jde sestavit jen kružnice  $C_5$ , protože na nesouvislý nemá dost hran.
- (7.5.11)\* 3
- (7.5.12)\* Třeba úplný bipartitní  $K_{3,4}$ .
- (7.5.13) Ano, ale jen otevřeným.
- (8.1.1) 1, ale jen 0 pro graf na jednom vrcholu.
- (8.1.2) 2
- (8.1.3) 5
- (8.1.4) 3, najdete příslušnou dvojici vrcholů?
- (8.2.1) Vyjde  $\begin{pmatrix} 0 & 1 & 2 & 1 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 1 \\ 1 & 2 & 1 & 0 \end{pmatrix}$ .
- (8.3.1) 5
- (8.3.2) Jsou to vrcholy ve zbylých dvou cípech hvězdy, do každého dalšího vrcholu je z nich vzdálenost nejvýše 4. Lépe to nejde.
- (8.5.1) Vlevo 3 a vpravo 2.
- (8.5.2) Vlevo 3 a vpravo 5.
- (8.5.3) 3. Ten prostřední vrchol má vzdálenost  $\leq 2$  do každého dalšího. Zbytek grafu je grafem krychle, kde největší vzdálenost 3 mají protilehlé dvojice vrcholů (ve smyslu geometrické krychle). Z těchto 4 dvojic má kratší spojení přes prostřední vrchol, takže zůstávají 3 dvojice.
- (8.5.4) 2 – jedná se o graf krychle, tak přidáme dvě úhlopříčky na jednu stěnu–čtverec. Pak budou vzdálenosti  $\leq 2$ . Naopak jedna hrana nestačí, protože po přidání libovolné hrany  $e$  stále najdeme dva protilehlé vrcholy krychle, které hraně  $e$  nepatří, a tyto dva vrcholy zůstanou ve vzdálenosti 3.
- (8.5.5)  $2 + 3 + 2 = 7$
- (8.5.6) 10. Vezmeme jeden vrchol  $v$ , ten má 3 sousedy a každý ze sousedů  $v$  má 2 další sousedy mimo  $v$ . To je dohromady 10 vrcholů a více jich už nemůže být, protože vzdálenosti jsou  $\leq 2$  od  $v$ . Nakreslete si takový graf!
- (8.5.7)\* 22, což je  $\lfloor 45/2 \rfloor$ , ohodnocení hran v pořadí 1, 2, 3, 4, 5, 7, 6, 8, 9.
- (8.5.8)\* Vzdálenost 14 mezi 7 a 5.
- (9.1.1) 3, jeden bez hran, jeden s jednou hranou a poslední strom s dvěma hranami.
- (9.1.2) 2, cesta  $P_3$  a “hvězda”  $K_{1,3}$ .
- (9.1.3)\* Nenačnete, to by bylo v rozporu s Důsledkem 9.5.

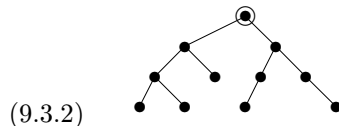


(9.2.2) Centra jsou vyznačená:



(9.2.3) Pokud centrum vyjde jeden vrchol, odebereme celkem 32 listů – za každou hranu jeden. Jinak odebereme jen 31 listů a jedna hrana zůstane jako centrum.

(9.3.1)  $((()((()()))))((()())())$



(9.4.1) 2004

(9.4.2)\* Je celkem  $5!/2 = 60$  koster, co jsou cestou  $P_4$ , dále jen 5 koster, co jsou “hvězdou”  $K_{1,4}$  a nakonec  $5 \cdot 4 \cdot 3 = 60$  koster, které mají vrchol stupně 3.

(9.5.1) 3

(9.5.2) Ano, 2005 vrcholů podle Věty 9.3.

(9.5.3)  $2009 - 1 - 1 - 1 - 1 = 2005$

(9.5.4) 11, počítejte hrany v každé komponentě – stromu.

(9.5.5) Graf není souvislý, 13 není spojeno s ničím. Není ani lesem, neboť 10, 14, 21, 15 tvoří kružnici.

(9.5.6) Vlevo B, vpravo A.

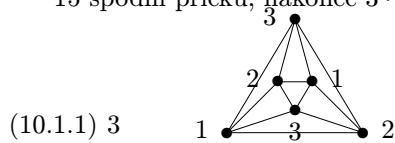
(9.5.7) 5

(9.5.8) 3 až 8 koster. Nejméně koster, 3, vznikne pokud nová hrana vytvoří trojúhelník. Nejvíce, 8, pokud strom byl cestou délky 7 a nová hrana ji uzavře do kružnice délky 8.

(9.5.9)  $\binom{n}{2} - n + 1$

(9.5.10)\* 4. Pro 3 vrcholy by každá kostra musela mít 2 hrany, ale  $2 + 2 = 4$  hrany mezi třemi vrcholy nelze mít. Na druhou stranu úplný graf  $K_4$  má takové dvě kostry.

(9.5.11)\* 56. Obvodová kružnice má 8 koster, dalších  $3 \cdot 5 = 15$  koster obsahuje vrchní příčku a 15 spodní příčku, nakonec  $3 \cdot 3 \cdot 2 = 18$  koster obsahuje obě příčky.



(10.1.2) Méně než 4 barvy nestačí, protože najdeme úplný podgraf na 4 vrcholech. Obarvení 4 barvami je snadné.

(10.1.3) 1 pokud má jeden vrchol a jinak 2 podle Věty 10.5 – stromy totiž nemají žádné kružnice.

(10.1.4) 3

(10.1.5)\* 102 barvami, postup je následovný: Odebereme jeden vrchol  $v$ , zbytek grafu rekurzivně obarvíme 102 barvami a v nejhorším případě dostanou sousedé vrcholu  $v$  101 různých barev. Proto i  $v$  můžeme korektně dobarvit.

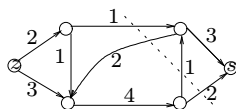
(10.2.1) Je to graf krychle.

(10.2.2)  $v = 12$ ,  $f = 20$ ,  $v + f - e = 2$ , tedy  $e = 12 + 20 - 2 = 30$ .

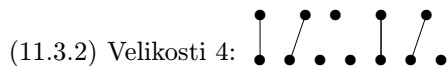
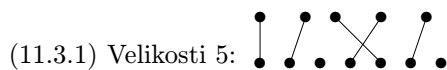
(10.2.3) Graf pravidelného osmistěnu.



- (10.2.4) 3  
 (10.3.1) Pro  $n = 1, 2, 3, 4$ .  
 (10.3.2) Rovinný jen  $B$ .  
 (10.3.3) 3 – takto z kružnice délky 6 vytvoříme graf  $K_{3,3}$ , nakreslete si to.  
 (10.5.1) 3  
 (10.5.2) 4, všimněte si, že graf obsahuje  $K_4$ .  
 (10.5.3) 3  
 (10.5.4) 1  
 (10.5.5)\* Buď 13, pokud vypustíme tři hrany jednoho trojúhelníku, nebo 12, pokud vypustíme tři disjunktní hrany.  
 (10.5.6) Ano, stačí dolní levé dva vrcholy překreslit nahoru.  
 (10.5.7) Ne, obsahuje podrozdělení  $K_{3,3}$ , najdete jej?  
 (10.5.8) 8  
 (10.5.9) 11  
 (10.5.10) Má 10 vrcholů podle Eulerova vztahu.  
 (11.2.1) Tok 5.  
 (11.2.2) Jen zdroj  $z$ .



(11.2.3) Řez velikosti 4 zde:



- (11.3.3) Ano, reprezentanti jsou zapsaní první:  $(1, 2, 3)$ ,  $(2, 1, 4)$ ,  $(3, 1, 4)$ ,  $(4, 2, 3)$ .  
 (11.3.4) Ne, všech podmnožin je  $\binom{5}{3} = 10$ , ale prvků k reprezentaci jen 5.  
 (11.4.1) 12  
 (11.4.2) 13  
 (11.4.3) 11  
 (11.4.4) 9  
 (11.4.5) Nemá – 6 z množin má sjednocení  $\{3, 4, 5, 6, 7\}$ , kde se najde jen 5 různých prvků pro reprezentanty.  
 (11.4.6) Nemá – 6 z množin má sjednocení  $\{1, 2, 3, 4, 9\}$ , kde se najde jen 5 různých prvků pro reprezentanty.

## Reference

- [1] P. Hliněný, Web stránky předmětu DIM (2004),  
<http://www.cs.vsb.cz/hlineny/vyuka/DIM.html>.
- [2] D. Knuth, The Art of Computer Programming, Volumes 1-3, Addison-Wesley Pub Co, 2nd edition 1998.
- [3] L. Kučera, Kombinatorické Algoritmy, SNTL, Praha, 1983.
- [4] L. Kučera, Algovize,  
<http://kam.mff.cuni.cz/~ludek/Algovision/Algovision.html>.
- [5] J. Matoušek a J. Nešetřil, Kapitoly z Diskrétní Matematiky, Karolinum, UK Praha, 2000.
- [6] J. Matoušek and J. Nešetřil, Invitation to Discrete Mathematics, Oxford University Press, 1998.
- [7] E. Ochodková, Grafové Algoritmy,  
<http://www.cs.vsb.cz/ochodkova/courses/gra/>.