Grafové algoritmy

2009/10, 1. termín

1. Maximální párování

Je dán neorientovaný graf $G = (V, E)$. Množina $M \subseteq E$ je *párování*, pokud žádné dvě různé hrany z M nemají společný vrchol. *Maximalita* párování se rozumí vzhledem k počtu hran.

Vrchol $v \in V$ je *volný* vzhledem k párování M, není-li koncem žádné z hran v M. Cesta $P = (v_0, v_1, \ldots, v_k)$, $k \geq 1$, v G je *střídavá* vzhledem k párování M, jsou-li vrcholy na ní po dvou různé a střídají-li se v ní hrany z M a $E \setminus M$. Taková cesta je *volná* vzhledem k M, jsou-li oba její konce volné. Její *alternací* dostáváme párování M', které se od M liší právě tím, že pro libovolnou hranu e na P je $e \in M$ právě když $e \notin M'$ a naopak.

a) Dokažte větu:

Věta. M je maximální párování v G právě když v G neexistuje volná cesta vzhledem k M.

Pozn. V důkazu netriviální implikace můžete uvážit dvě párovaní M a M' a diskutovat jak vypadají souvislé komponenty grafu $(V, M \cup M')$.

Ve zbytku pojednání je graf G *bipartitní*, tj. existují neprázdné disjunktní množiny vrcholů X, Y dávající ve sjednocení celé V a takové, že libovolná hrana z E ma jeden konec v X a druhý v Y. Fixujme taková X, Y.

V následujícím algoritmu je
$Match[y] = x$, je-li $\{x, y\} \in M$ a
$Match[y] = 0$, je-li y volný vzhledem k M.

Jeho podstatou je hledání volných cest z $u \in X$ pomocí BFS(G). Volné cesty z $v \in Y$ nemusíme uvažovat (stačí je opačně orientovat). Pro cestu $(u, v, u', v', u``, v'', \ldots)$ klademe $Prev[v] = u,\ Prev[v'] = u', \ldots$. Navštívené vrcholy z X máme v poli $Q[1], Q[2], \ldots$ a ty z Y množině N.

b) V uvedených algoritmech doplňte řádky 18 resp. 4.

```
MaxMatching(G)
 1  for i ← 1 to n
 2      Match[i] ← 0
 3  for u ∈ X
 4      Q[1] ← u, Qsize ← 1
 5      N ← ∅
 6      for i ← 1 to n
 7          Prev[i] ← 0
 8      k ← 1
 9      repeat
10          x ← Q[k]
11          for y ∈ Adj[x]
12              if y ∉ N
13                  přidej y do N
14                  Prev[y] ← x
15                  if y je volný
16                      Alternuj(y)
17                      goto 1
18                  přidej ......... do Q
19          k ← k + 1
20      until k > Qsize
21      odstraň z grafu vrcholy z Q a N včetně incidentních hran
22      1:
```

```
Alternuj(y)
1  repeat
2      w ← Prev[y]
3      Match[y] ← w
4      ...
5      Match[w] ← y
6      y ← v
7  until y = 0
```

c) Doplňte důkaz věty :

Věta. Množina $M = \{ \{x, Match[x]\} \mid x \in X,\ Match[x] \neq 0 \}$ je maximálním párováním grafu G.

Důkaz. Pokud žádný z X nezůstane volný, máme maximální párování. Pokud z $u \in X$ najdeme volnou cestu, alternujeme ji. Pokud z $u \in X$ existuje volná cesta, najdeme ji, neboť před ř. 21 máme v Q vrcholy právě vrcholy z X dosažitelné z u pomocí

.... začínající hranou ....

a v množině N ....

Pokud volná cesta z u neexistuje, máme před ř. 21 :

(i) $|Q| = |N| +$ ....

(ii) každý vrchol z N je zpárován s ....

(iii) neexistuje hrana mezi Q a $Y \setminus N$.

Zbývá ukázat, že vrcholy z Q a N a s nimi incidentní hrany je možné „beztrestně“ odstranit.

I kdyby jsme je neodstranili, nemůže v budoucnu vést volná cesta z $u' \in Q$, $u' \neq u$, neboť

....

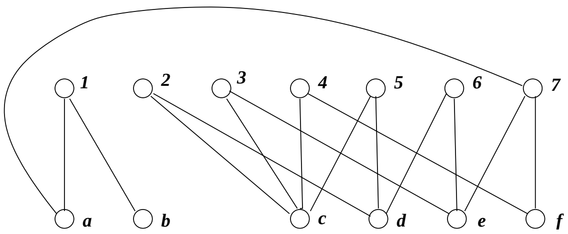
Rovněž, kdyby taková cesta vedla z $u' \in X \setminus Q$ a nezůstala celá v $X \setminus Q$ a $Y \setminus N$, navštívila by nejprve nějaké $v \in N$ před tím než by mohla do Q kvůli ....
Pak by pokračovala do nějakého

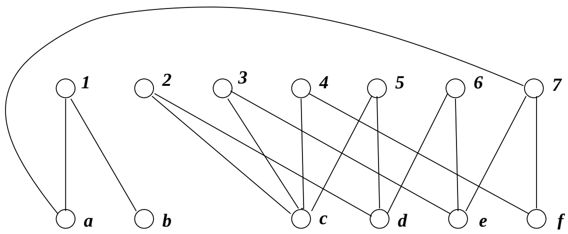
....
a poté do nějakého

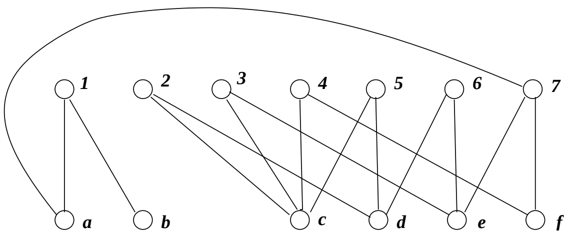
...., atd.,
a nemohla by skončit v nějakém volném vrcholu kvůli

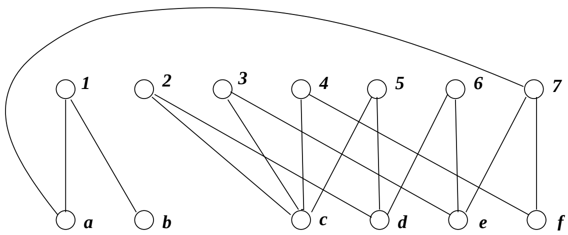
....
- SPOR

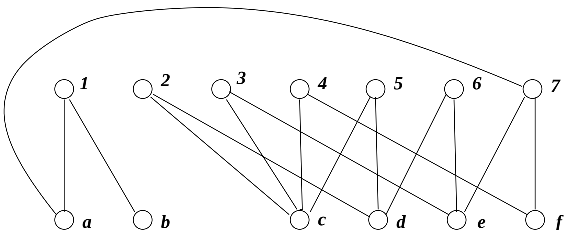
d) Do přiložených diagramů vyznačte průběch algoritmu. Hrany aktuálního párování značte červeně, orientované hrany $(y, Prev[y])$ značte modře a odstraněné vrcholy a hrany budou černě (pozor : odstraněná hrana může být v párování). Nový obrázek kreslete, kdykoliv proběhne některý z řádků 7,16 (vyznačte jen změny),20. Můžete vynechat malování pro $u = 1, \ldots, 5$ a začít s $u = 6$. Seznamy sousedů jsou seřazeny podle velikosti či abecedy. Rovněz výběr u probíhá podle velikosti. Vyznačujte též $Q[1], Q[2], \ldots, Q[Qsize]$ a vrcholy $y_1, y_2, \ldots$ z řádku 13 (ty se budují pro nové u znovu).

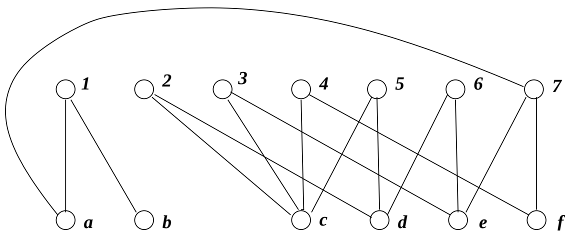

1 2 3 4 5 6 7
a b c d e f

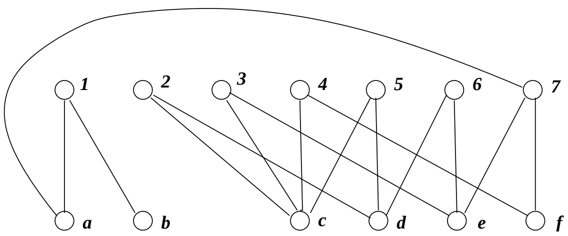

1 2 3 4 5 6 7
a b c d e f

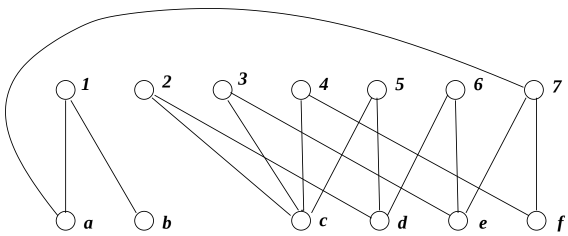

1 2 3 4 5 6 7
a b c d e f

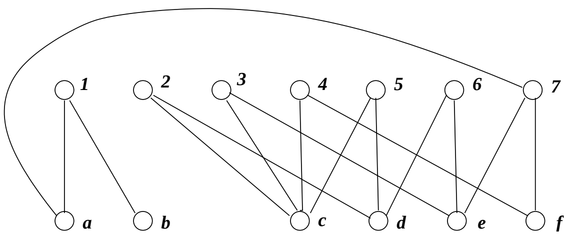

1 2 3 4 5 6 7
a b c d e f

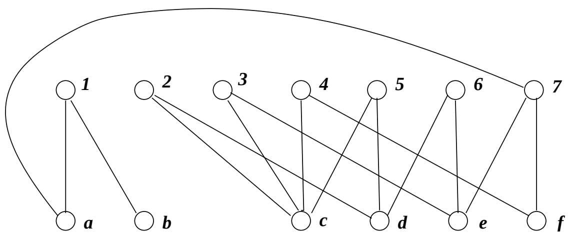

1 2 3 4 5 6 7
a b c d e f


1 2 3 4 5 6 7
a b c d e f


1 2 3 4 5 6 7
a b c d e f


1 2 3 4 5 6 7
a b c d e f


1 2 3 4 5 6 7
a b c d e f


1 2 3 4 5 6 7
a b c d e f

Dijkstrův algoritmus

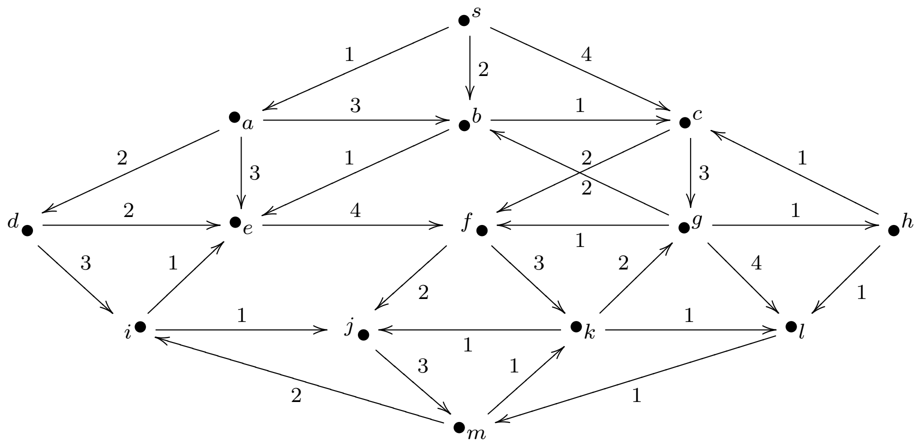
Kontrolní otázky

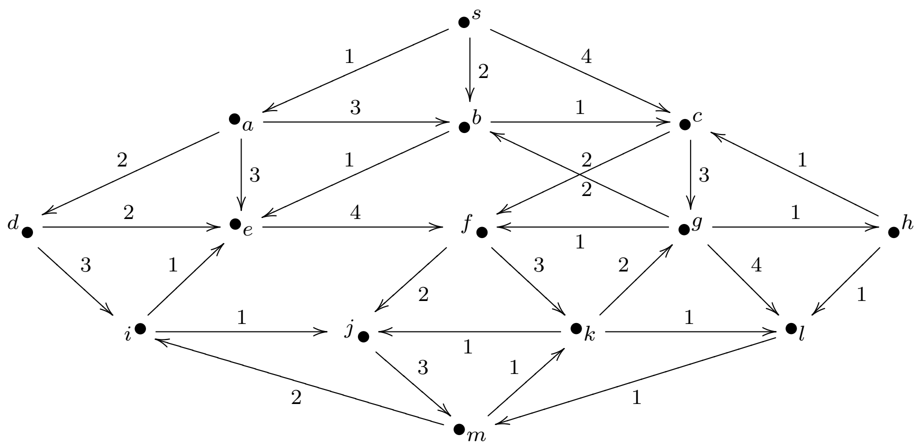
a) Napište jednoduchý příklad orientovaného hranově ohodnoceného grafu G a jeho vrcholu s takového, že Dijkstrův algoritmus při hledání nejkratších cest z vrcholu s v grafu G nevypočítá správný výsledek. U každého vrcholu uveďte jak správnou hodnotu nejkratší cesty z s do tohoto vrcholu, tak výsledek vypočítaný Dijkstrovým algoritmem.

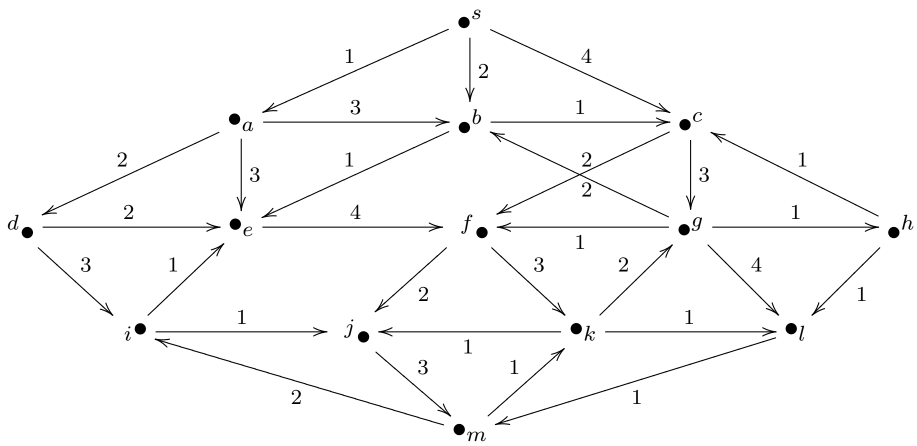
b) Napište jednoduchý příklad orientovaného hranově ohodnoceného grafu G a jeho vrcholu s takového, že: Graf G má alespoň 4 vrcholy, alespoň jedna jeho hrana má kladné a alespoň jedna hrana záporné ohodnocení a Dijkstrův algoritmus při hledání nejkratších cest z vrcholu s v grafu G vypočítá správný výsledek.

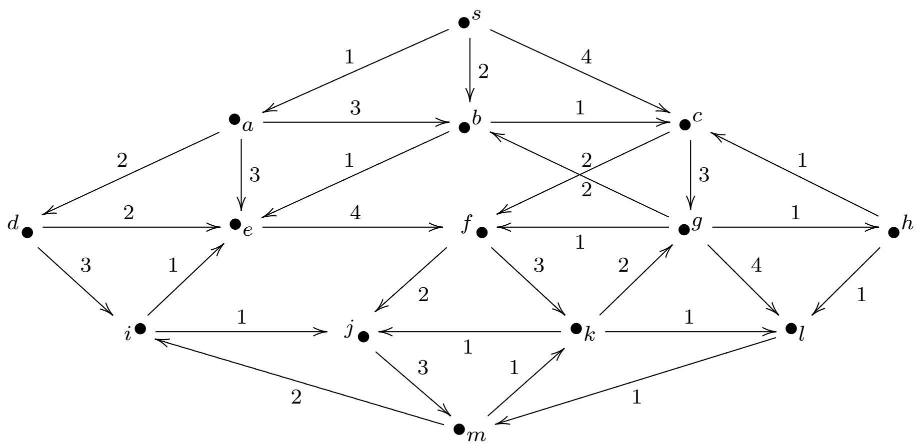
Výpočet

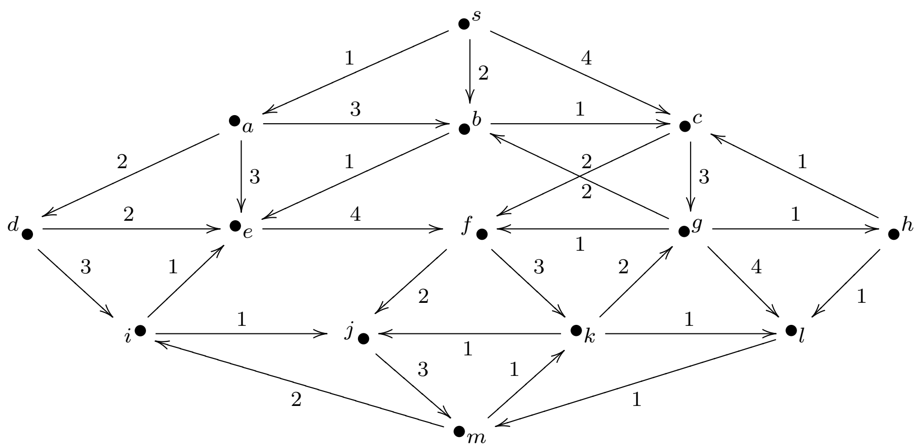
Pomocí Dijkstrova algoritmu nalezněte nejkratší cesty (a jejich délky) z vrcholu s do všech vrcholů zadaného grafu. Aby byl výpočet přehlednější, tak pokaždé, kdy je nějaký ukazatel $\pi[v]$ změněn z nějakého uzlu (tj. z hodnoty různé od *nil*) na jiný uzel, použijte k dalšímu výpočtu nový níže uvedený diagram. Vždy vyznačte aktuální hodnoty $d[v]$ u všech vrcholů v, již zpracované vrcholy, pro které je hodnota vypočítána správně a hrany stromu indukovaného ukazateli π (nejlépe tak, jak je to v ilustračním příkladu v brožurce). Je-li v některé iteraci na výběr z více uzlů, vybírejte v abecedním pořadí.











Grafové algoritmy

2009/10, 2. termín

1. Maximální cesta

Je dán souvislý neorientovaný graf $G = (V, E)$. Terminologie a značení :
$n = |V|$, $m = |E|$,
$deg(v)$ značí stupeň vrcholu $v \in V$,
místo $\{u, v\}$ píšeme stručně uv,
$(v_0, v_1, \ldots, v_k)$ je *cesta*, je-li $k \geq 0$, $v_0, v_1, \ldots, v_k$ jsou po dvou různé prvky z V a $v_0v_1, \ldots, v_{k-1}v_k \in E$,
$(v_0, v_1, \ldots, v_k)$ je *kružnice*, je-li $k \geq 3$, $v_0 = v_k$, $v_0, v_1, \ldots, v_{k-1}$ jsou po dvou různé prvky z V a $v_0v_1, \ldots, v_{k-1}v_k \in E$,
Ta je *hamiltonovská*, je-li $k = n$.

V algoritmu se střídají čtyři fáze :
1. Cestu $P = (u, \ldots, v)$ rozšiřujeme nadoraz doleva (začíná se s $P = (x)$).
2. Cestu $P = (u, \ldots, v)$ rozšiřujeme nadoraz doprava.
3. Máme-li na cestě $P = (u, \ldots, v)$ vrchol w takový, že $vw, uw^+ \in E$, $P = (u, \ldots, w, w^+, \ldots, v)$ (může být $w = u$ či $w^+ = v$), modifikujeme P na kružnici C.
4. Kružnici C modifikujeme na cestu, která má o jednu hranu více než původní P.

Nechť $x \in V$ je pevně vybraný.

a) V následujícím algoritmu doplňte řádky 6 a 12.

LongPath(G, x)

1 $u \leftarrow x,\ v \leftarrow x,\ P \leftarrow (x)$
2 **repeat**
3   **while** existuje $w \in V \setminus P$ splňující $wu \in E$
4   přidej w zleva k P, $u \leftarrow w$
5   **while** existuje $w \in V \setminus P$ splňující $wv \in E$
6   přidej w zprava k P, $v \leftarrow w$
7   **for** w na P
8   **if** $wv \in E$ and $w^+u \in E$
9             $C \leftarrow P + vw - ww^+ + uw^+$
10   **if** C je hamiltonovská
11   **goto 1**
12   najdi $z \in C,\ y \notin C,\ zy \in E$
13   modifikuj $C + zy$ na cestu P z y do z^+
14             $u \leftarrow y,\ v \leftarrow z^+$
15 **until** žádná změna v průběhu tohoto cyklu
16 **1:**

Složitost : předpokládejme, že v konstantním čase dokážeme zjistit, zda dané $w \in V$ je na P či na C a pro dané $p, q \in V$ zda $pq \in E$ či nikoliv.

b) Doplňte úvahy v následujícím odhadu složitosti :

Fáze 1 a 2. Množina vrcholů na P .... Proto když jsme jednou použili nebo odmítli jistou hranu

.... Tedy celkem ....

pro všechny průběhy fází 1 a 2.

Fáze 3. Na P procházíme možná w - těch je .... Při úspěchu převracíme část P - to vezme .... Takových P máme .... Tedy celkem všechny fáze 3 vezmou ....

Fáze 4. Výběr z pro danou fázi 4 vezme .... Celkem výběr z pro všechny fáze 4 vezme ....

Zjištění, zda dané z je vhodné vezme .... Pokud jsme z nějakého z neuspěli, nepodaří se nám to ani nikdy v budoucnu. Tedy pro všechny fáze 4 a pro všechna z potřebujeme

.... Všechny fáze 4 vezmou ....

Celý algoritmus potřebuje ....

c) Doplňte důkaz věty :

Věta. Nechť pro každé $u, v \in V,\ u \neq v,\ uv \notin E$ máme $deg(u) + deg(v) \geq n$. Pak náš algotitmus najde hamiltonovskou kružnici.

Důkaz. Připustme, že jsme nenalezli hamiltonovskou kružnici a že jsme skončili s cestou $P = (u, \ldots, v)$ s, řekněme s k hranami. P nemůže být prodloužena ze sých konců a proto hrany z u a v vedou

....

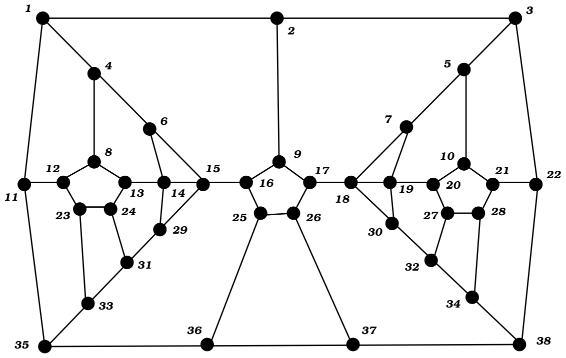
Dále pro w na P máme : $vw \in E$ dá ....

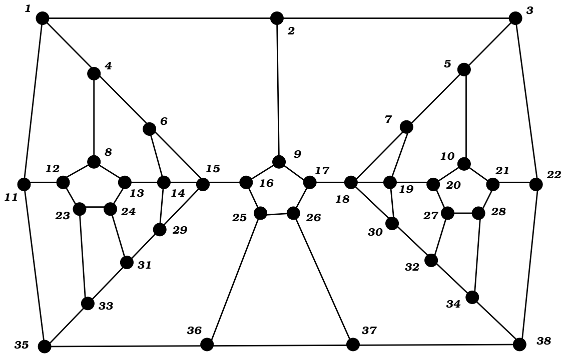
Tedy hrany z u mohou vést do maximálně ....

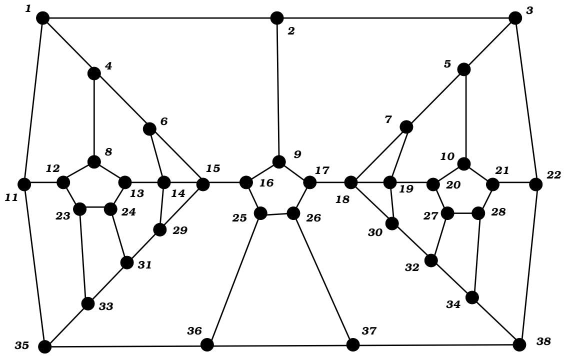
vrcholů. Tedy $deg(u) \leq$ ....,

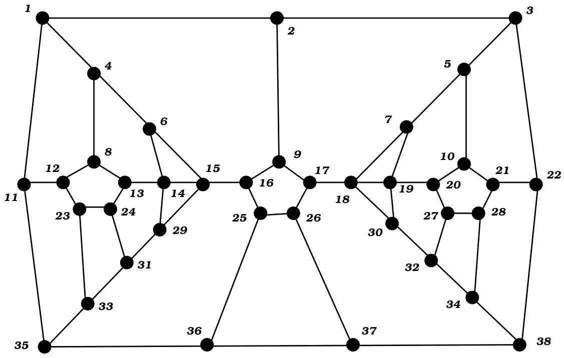
což dá $deg(u) + deg(v) \leq$ ....
- SPOR

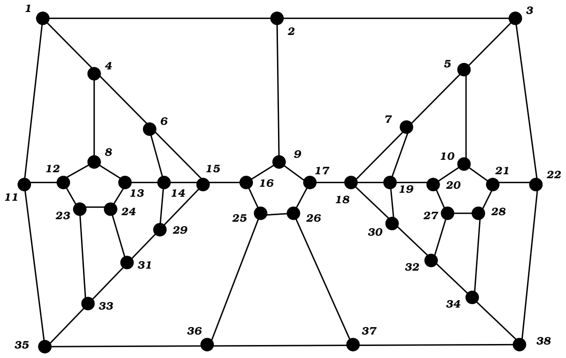
d) Do přiložených diagramů vyznačte průběch algoritmu. Dáno $x = 12$. Výběr w, z, y na ř. 3, 5, 12 respektuje pořadí $1 < 2 < \ldots.$ Výběr w na ř. 7 probíhá po P zleva doprava. Nový stav zachyťte vždy po provedení 5-6, 9, 13. P je zelená, C je červená.

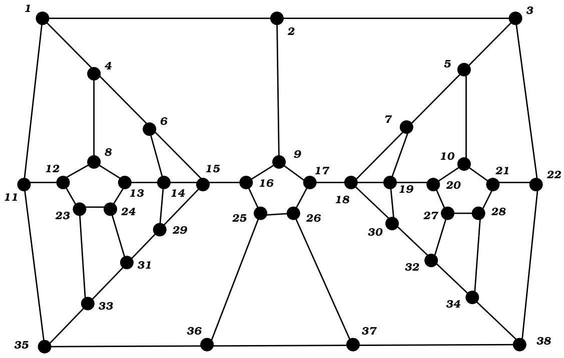

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38

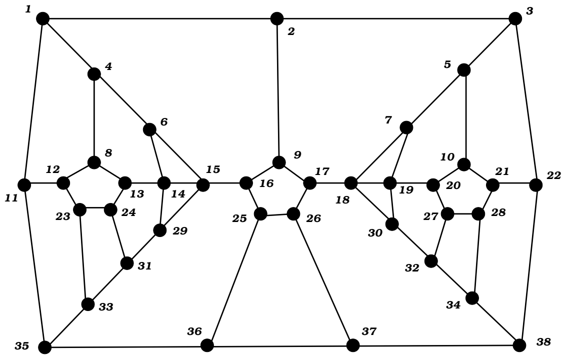

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38

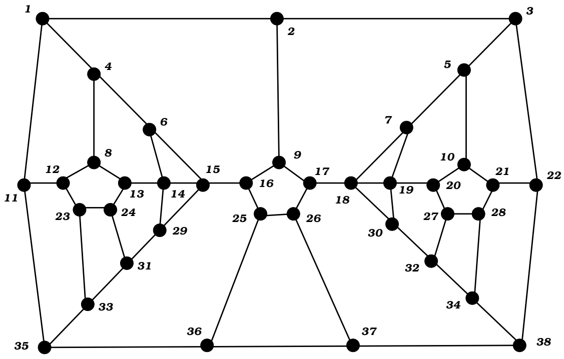

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38

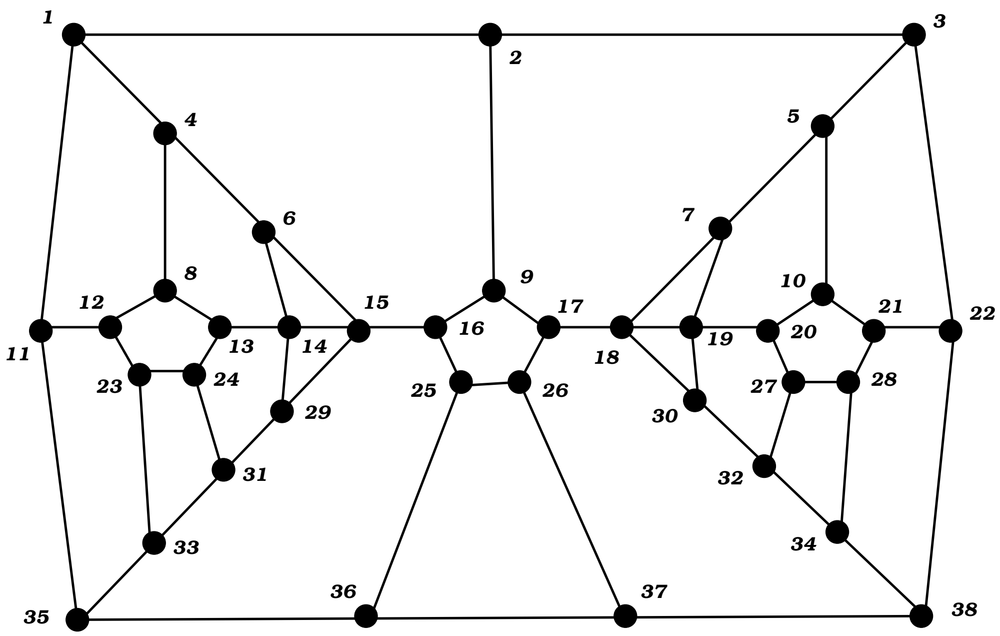

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38

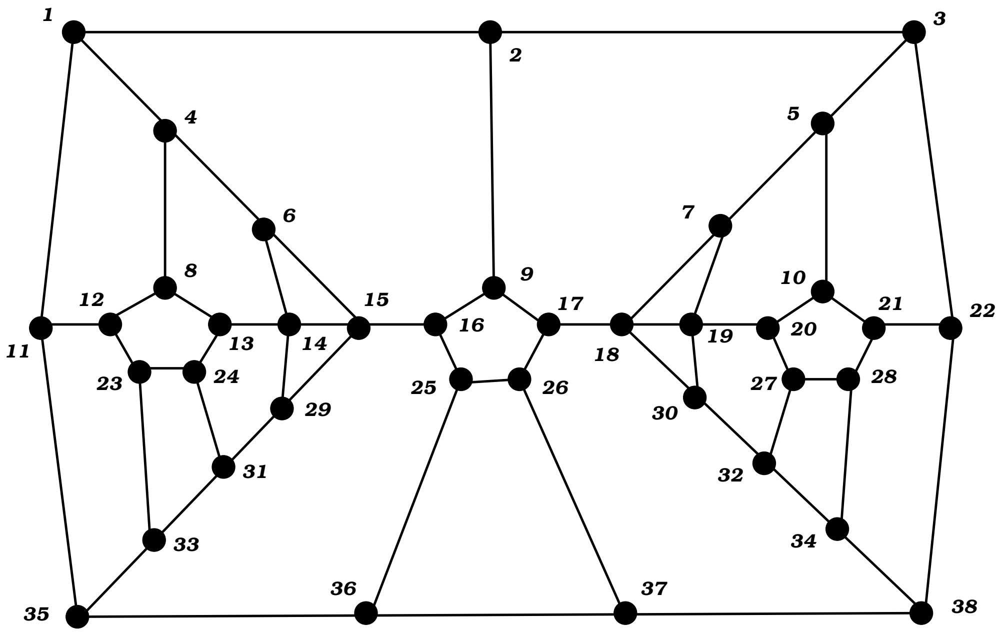

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38

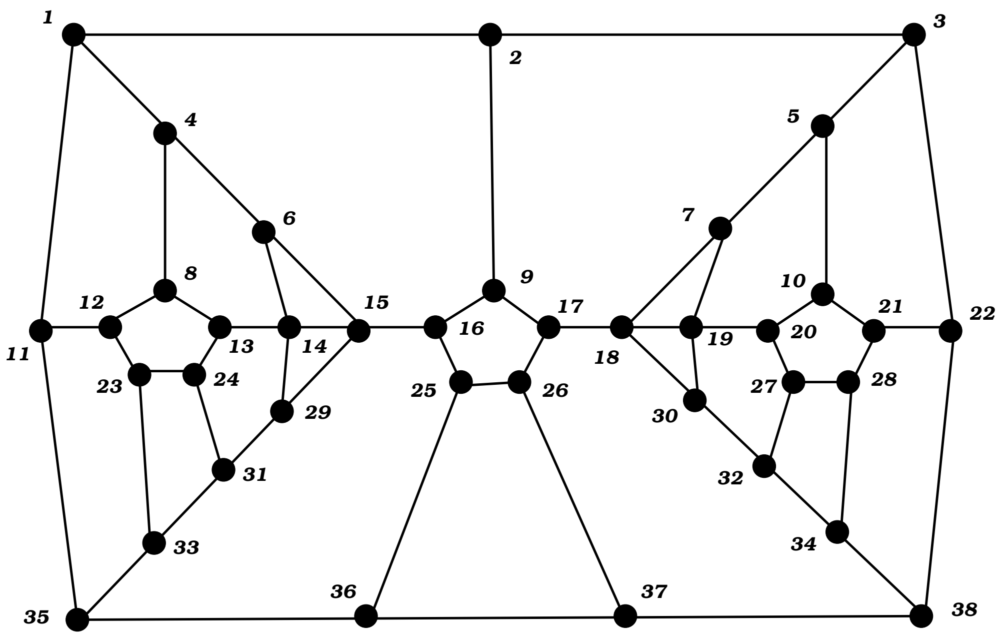

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38

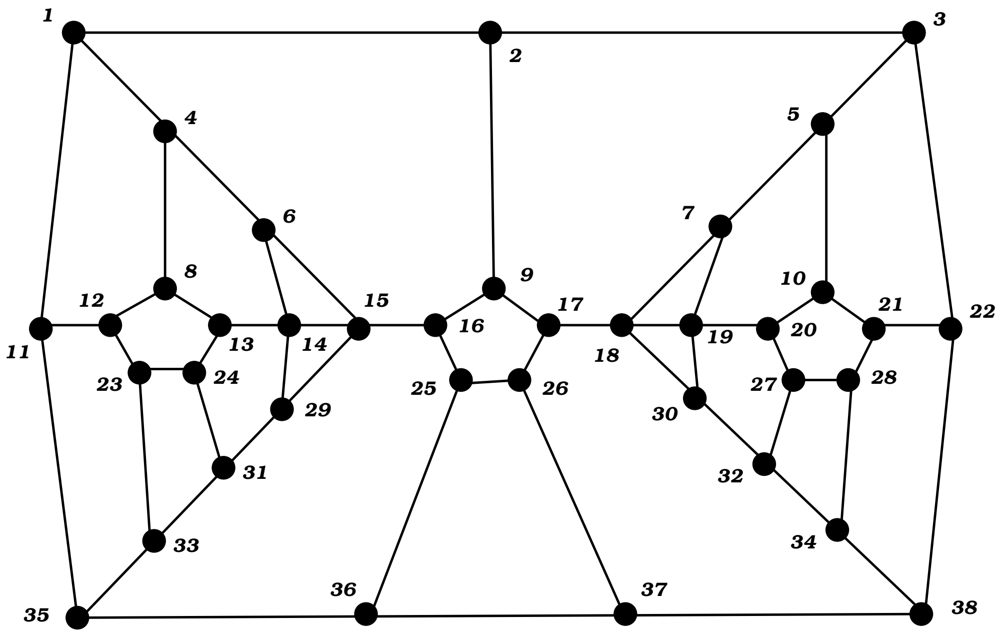

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38


1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38

Minimální kostry

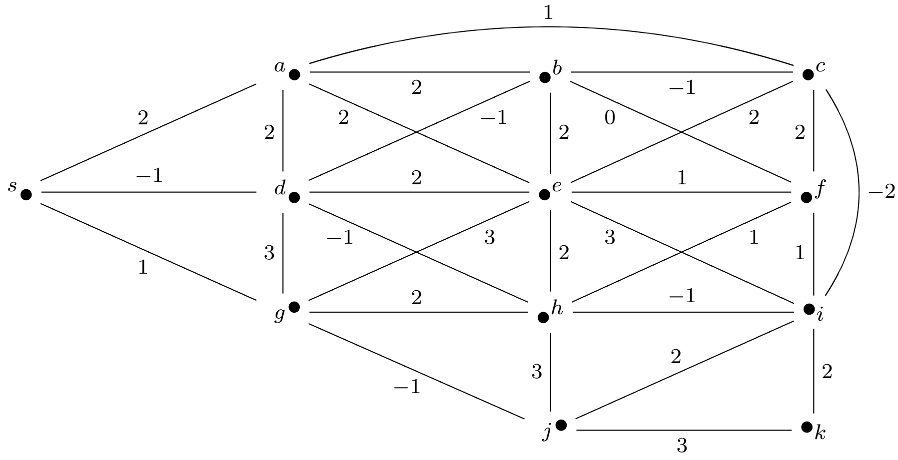
Kontrolní otázky

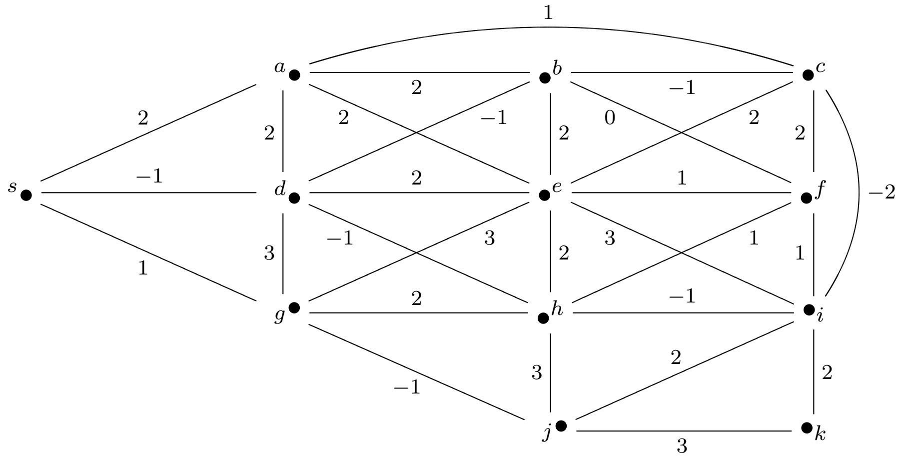
a) Napište jednoduchý příklad neorientovaného souvislého hranově ohodnoceného grafu s 5 vrcholy, který má přesně 6 různých minimálních koster.

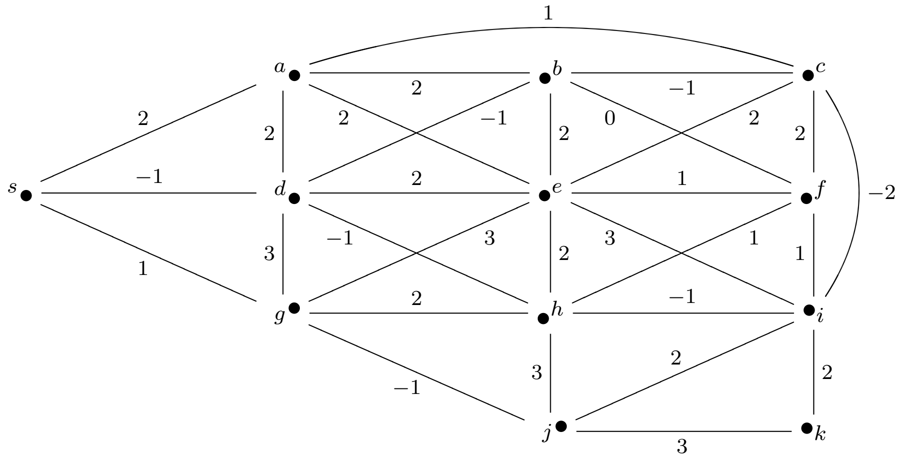
b) Nechť $n \geq 5$ je libovolné, ale nadále pevně zvolené přirozené číslo. Napište, kolik různých minimálních koster má graf $G = (V, E, w)$, kde $V = \{1, \ldots, n\}$, $E = \{\{i, i+1\} \mid 1 \leq i \leq n-1\} \cup \{\{n, 1\}, \{1, 3\}\}$ a ohodnocení $w(\{x, y\}) = 1$ pro libovolnou hranu $\{x, y\} \in E$.

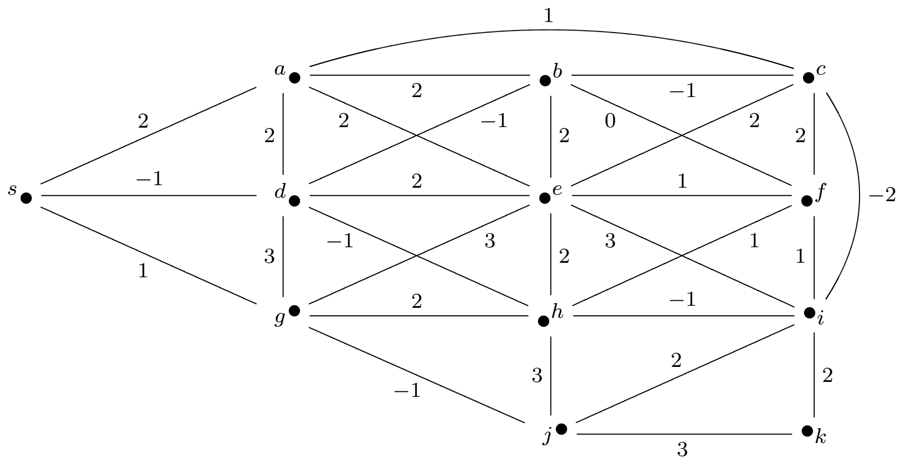
Výpočet

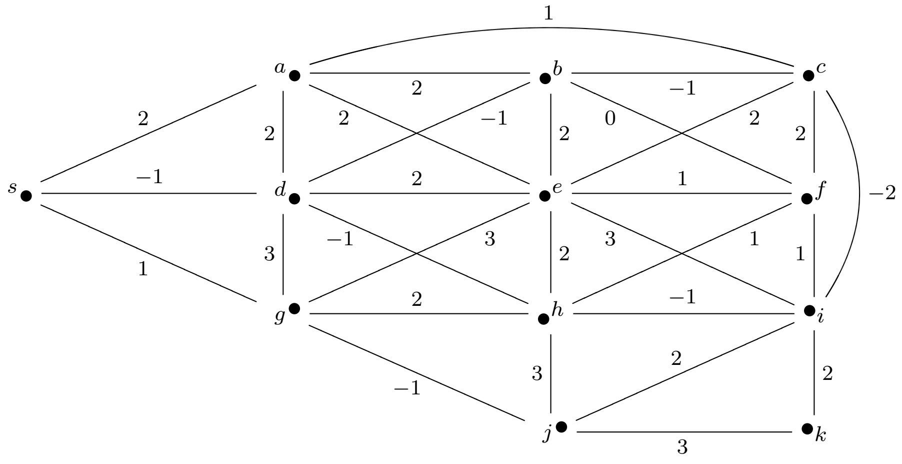
Pomocí **Primova** algoritmu nalezněte nějakou minimální kostru zadaného grafu. Aby byl výpočet přehlednější, tak pokaždé, kdy je nějaký ukazatel $\pi[v]$ změněn z nějakéjo uzlu (tj. z hodnoty různé od *nil*) na jiný uzel, použijte k dalšímu výpočtu nový níže uvedený diagram. Vždy vyznačte při tomto přechodu aktuální hodnoty $key[v]$ u všech vrcholů v, hrany které jsou již součástí minimální kostry a hrany, které jsou ve stromě indukovaném ukazateli π, ale nejsou dosud součástí minimální kostry (tj. jeden jejich vrchol dosud leží ve frontě Q). Kvůli jednoznačnosti začněte výpočet z vrcholu s a je-li v některé iteraci na výběr z více vrcholů, vybírejte v abecedním pořadí.

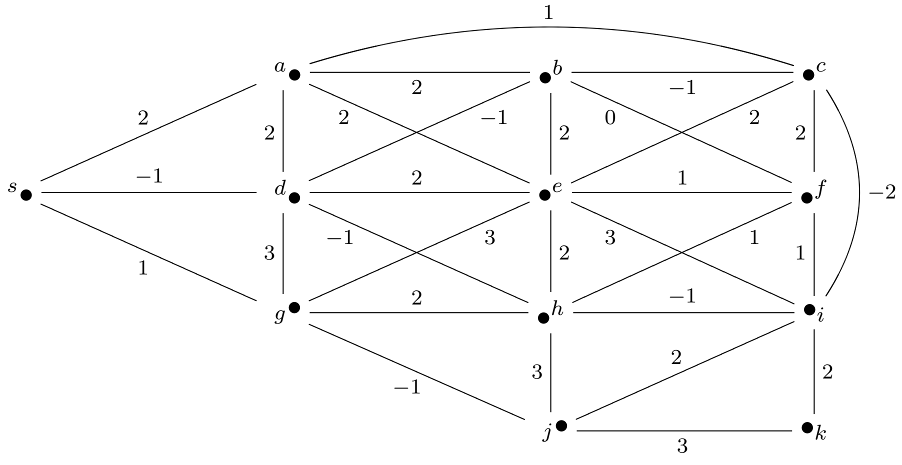

1
a b c
2 −1
2 2 −1 2 0 2 2
s d e f −2
−1 2 1
3 −1 3 2 3 1 1
1
g h i
2 −1
3 2 2
−1
j k
3


1
a b c
2 −1
2 2 −1 2 0 2 2
s d e f −2
−1 2 1
3 −1 3 2 3 1 1
1
g h i
2 −1
3 2 2
−1
j k
3


1
a b c
2 −1
2 2 −1 2 0 2 2
s d e f −2
−1 2 1
3 −1 3 2 3 1 1
1
g h i
2 −1
3 2 2
−1
j k
3


1
a
b
c
2
−1
2
2
−1
2
0
2
2
s
−1
d
2
e
1
f
−2
3
−1
3
2
3
1
1
1
g
2
h
−1
i
2
3
2
−1
j
3
k


1
a
b
c
2
−1
2
2
−1
2
0
2
2
s
−1
d
2
e
1
f
−2
3
−1
3
2
3
1
1
1
g
2
h
−1
i
2
3
2
−1
j
3
k


1
a
b
c
2
−1
2
2
−1
2
0
2
2
s
−1
d
2
e
1
f
−2
3
−1
3
2
3
1
1
1
g
2
h
−1
i
2
3
2
−1
j
3
k

Grafové algoritmy
2009/10, 3. termín

1. Bloky souvislého neorientovaného grafu

Je dán souvislý neorientovaný graf $G = (V, E)$. Terminologie a značení :
místo $\{u, v\}$ píšeme stručně uv,

$(v_0, v_1, \ldots, v_k)$ je *cesta*, je-li $k \geq 0$, $v_0, v_1, \ldots, v_k$ jsou po dvou různé prvky z V a $v_0v_1, \ldots, v_{k-1}v_k \in E$; často též píšeme $(v_0v_1, \ldots, v_{k-1}v_k)$.

$(v_0, v_1, \ldots, v_k)$ je *kružnice*, je-li $k \geq 3$, $v_0 = v_k$, $v_0, v_1, \ldots, v_{k-1}$ jsou po dvou různé prvky z V a $v_0v_1, \ldots, v_{k-1}v_k \in E$,

Hrana e je *most*, když po jejím odstranění graf přestává být souvislý. Nechť M značí množinu všech mostů.

Na množině $E \setminus M$ definujeme relaci $\sim$ vztahem $e \sim f$ právě když hrany e, f leží na nějaké společné kružnici.

a) Doplňte důkaz.
Lemma 1. Relace $\sim$ na množině $E \setminus M$ je relací ekvivalence.
Důkaz.

Bloky v G jsou následující podgrafy grafu G :
- most se svými koncovými vrcholy,
- pro $e \in E \setminus M$, třída $e\sim$ spolu s konci jednotlivých hran.

Neprázdná množina hran B spolu s jejich koncovými vrcholy je *bisouvislá*, leží-li její libovolné dvě hrany na společné kružnici nebo jedná-li se o jedinou hranu.

b) Formulujte vztah mezi bisouvislými podgrafy a bloky (vše netriviální) a vaše tvrzení dokažte.
Lemma 2.

Následující algoritmus hledá bloky grafu G; přitom

C_{uv} je (jediná) kružnice vzniklá přidáním hrany uv ke stromu T – tzv. fundamentální kružnice pro hranu uv,

B_{uv} má být blok obsahucící hranu uv,

Q je fronta vrcholů - je to jako v BFS,

T je BFS-strom.

c) V algoritmu doplňte řádky 12 a 15.

Blocks(G)

1 **for** $uv \in E$
2 $\quad B_{uv} \leftarrow uv$
3 vyber $x \in V$
4 $Q \leftarrow x$
5 **repeat**
6 $\quad u \leftarrow head\ Q$
7 $\quad$ **for** $v \in\ Adj\ u$
8 $\quad\quad$ **if** $v \notin Q$
9 $\quad\quad\quad$ přidej uv do T, přidej v na konec Q
10 $\quad\quad$ **else for** $xy \in C_{uv}$
11 $\quad\quad\quad\quad B_{uv} \leftarrow B_{uv} \cup B_{xy}$
12 $\quad\quad\quad$ **for** xy ....
13 $\quad\quad\quad\quad B_{xy} \leftarrow B_{uv}$
14 $\quad\quad$ odstraň u z $Adj\ v$
15 $\quad$ odstraň u ....
16 **until** $Q = \emptyset$

d) Doplňte důkaz tvrzení :

Lemma 3. Po každé iteraci cyklu 7-14 je každé B_{uv} bisouvislé.

Důkaz. Indukcí vzhledem k počtu k již provedených cyklů.

$k = 0$. Libovolné $B_{uv} =$ .... a tedy ....

Indukční krok : 1. Nechť $v \notin Q$. Pak ....

2. $v \in Q$. Vytvoří se kružnice $u, v = x_0, x_1, \ldots, x_k = u$, $x_0x_1, \ldots, x_{k-1}x_k \in T$.

Pak $B =$ .... je novou hodnotou pro B_{xy}, $xy \in$ ....

Nechť $i \in \{1, \ldots, k\}$, $ab \in B_{x_{i-1}x_i}$. Pro $ab = x_{i-1}x_i$ máme ab a uv na kružnici v B.

Pro $ab \neq x_{i-1}x_i$ máme ab a $x_{i-1}x_i$ .... podle ....

a podle .... máme ab a uv na kružnici v B.

Tranzitivita $\sim$ dá zbytek.

e) Doplňte důkaz tvrzení :

Věta. Po skončení algoritmu je pro každé $uv \in E$ graf B_{uv} blokem obsahujícím hranu uv.

Důkaz. Připustme, že B_{uv} není blok. Nechť B je blok obsahující hranu uv. Máme $B_{uv} \subsetneqq B$. Nechť $xy \in B \setminus B_{uv}$. Existuje kružnice C v B obsahující

....

Ukážeme, že $xy \in B_{uv}$ a to bude spor.

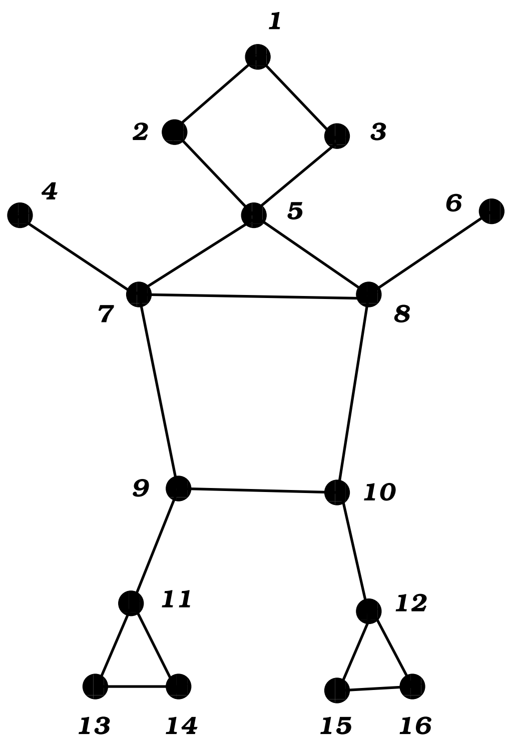
....

....

....

Můžete použít : Nechť hrany z $E \setminus T$ na C jsou $e_1, \ldots, e_k$. Nechť příslušné fundamentální kružnice jsou $C_1, \ldots, C_k$. Kružnice považujeme za sousední, mají-li společnou hranu. Lze ukázat, že takovýto graf (s vrcholy $C_1, \ldots, C_k$) je souvislý.

f) Demonstrujte algoritmus na panáčkovi níže. Uveďte tabulku, v níž řádky odpovídají změnám Q. Do prvního sloupce píšeme aktuální Q, pokud se něco přidávalo do T, uveďte to do 2. sloupce. Do 3. sloupce píšeme změněné C_{uv} a B_{uv}. Máme $x = 1$ a seznamy sousedů jsou uspořádány podle velikosti označení sousedů od nejmenšího po největší. V diagramu znázorněte bloky.



Silně souvislé komponenty

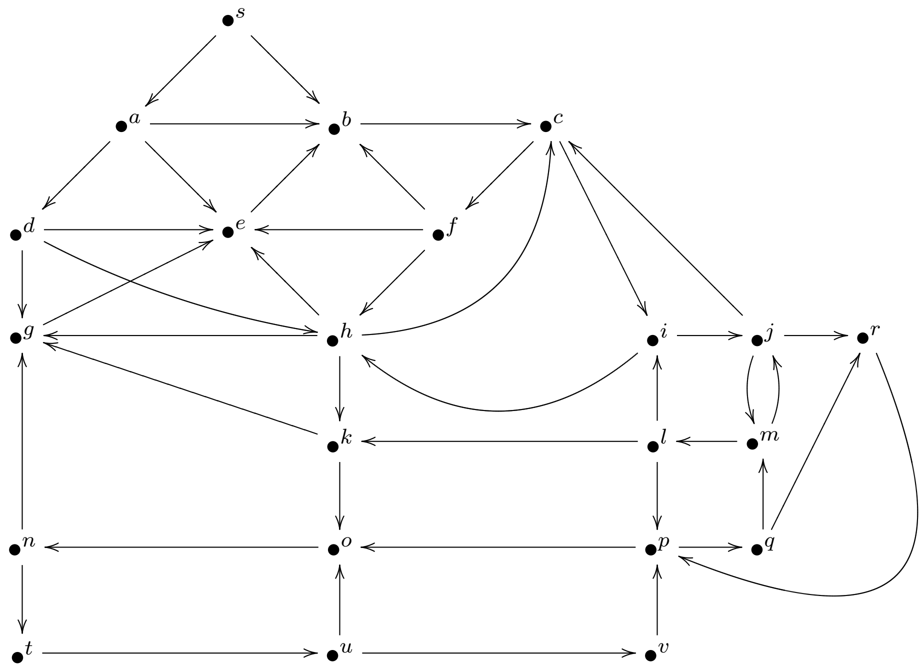
Kontrolní otázky

a) Napište jednoduchý příklad orientovaného grafu, který má 5 silně souvislých komponent, avšak změnou orientace jedné hrany (tzn. otočením jedné šipky) klesne počet silně souvislých komponent na 1. Příslušnou hranu v grafu vyznačte.

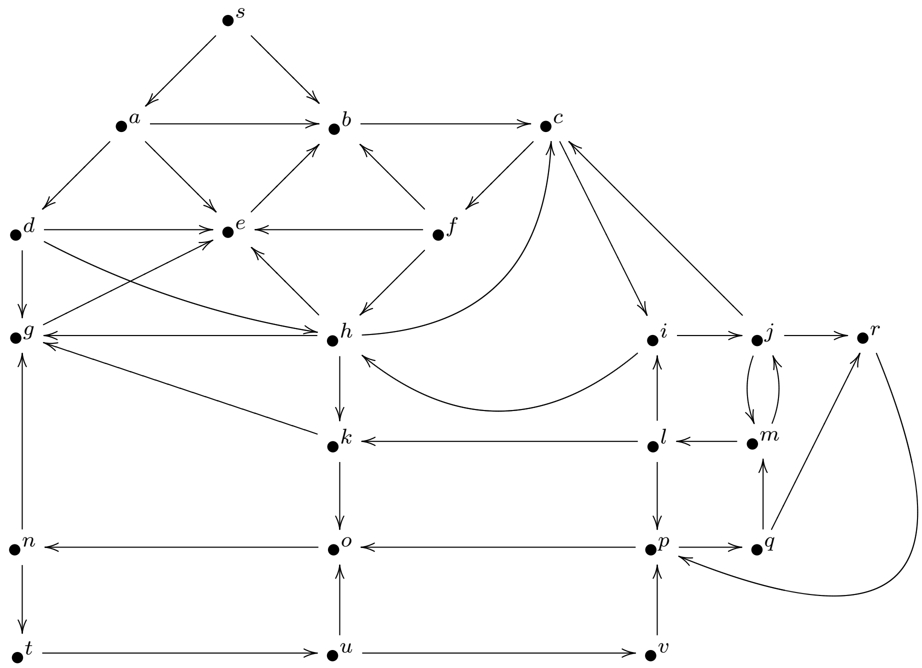
b) Napište jednoduchý příklad orientovaného grafu, který má 5 silně souvislých komponent, avšak změnou grafu na neorientovaný graf (tzn. že ke každé hraně přidáme opačně orientavanou hranu, pokud tato v grafu neexistuje) získáme graf, který má 3 (silně) souvislé komponenty.

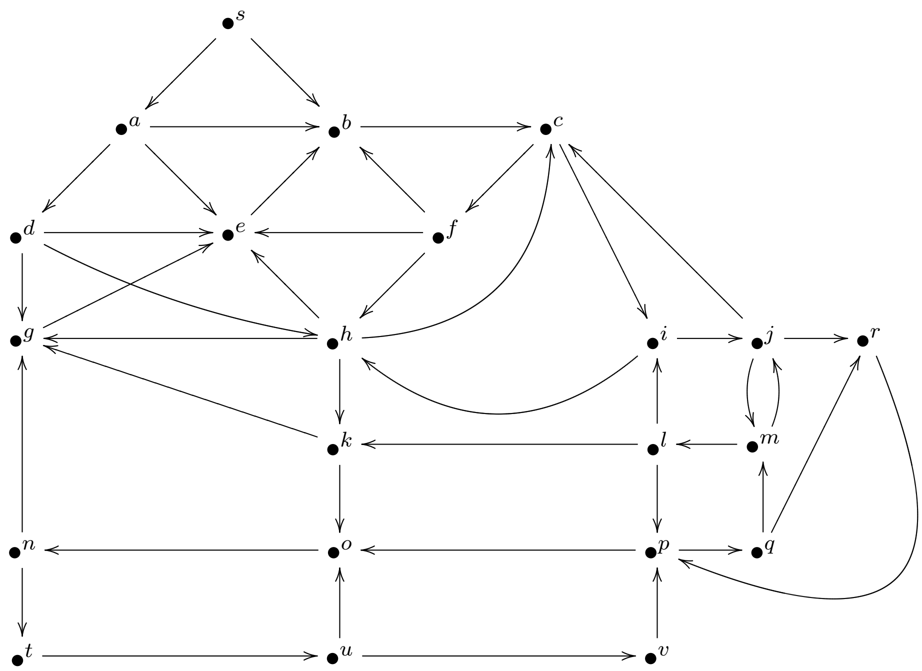
Výpočet

Na následujícím grafu nejprve proveďte prohledání do hloubky (DFS). Kvůli jednoznačnosti začněte prohledávat z vrcholu s, pokud máte v některém kroku na výběr, rozhodujte se dle abecedy (tzn. seznamy sousedů jsou uspořádány podle abecedy). Vyznačte hrany patřící do DFS stromu a ke každému vrcholu x napište hodnoty $d[x]$ a $f[x]$ (tzv. *discovery* a *finishing time*).



Nyní vhodným prohledáním transponovaného grafu najděte silně souvislé komponenty původního grafu. Opět platí, že transponovaný graf má seznamy následníků seřazené dle abecedy. K dispozici máte diagram původního grafu, nezapomeňte tedy, že tentokrát procházíme graf proti směru šipek. Opět vyznačte hrany patřící do DFS stromu, hodnoty $d[x]$ a $f[x]$ pro všechny vrcholy x a samozřejmě též nalezené silně souvislé komponenty.


s
a
b
c
d
e
f
g
h
i
j
r
k
l
m
n
o
p
q
t
u
v


s
a
b
c
d
e
f
g
h
i
j
r
k
l
m
n
o
p
q
t
u
v