

Diskrétní matematika – 4. týden

Elementární teorie čísel – Primitivní kořeny

Lukáš Vokřínek

Masarykova univerzita
Fakulta informatiky

podzim 2022

Obsah přednášky

- 1 Řád čísla, primitivní kořeny
- 2 Diskrétní logaritmus
- 3 Kvadratické zbytky a nezbytky
- 4 Výpočetní aspekty teorie čísel

Doporučené zdroje

- Jan Slovák, Martin Panák, Michal Bulant
Matematika drsně a svižně, e-text na
www.math.muni.cz/Matematika_drsne_svizne.

Doporučené zdroje

- Jan Slovák, Martin Panák, Michal Bulant
Matematika drsně a svižně, e-text na
www.math.muni.cz/Matematika_drsne_svizne.
- Michal Bulant, výukový text k přednášce **Elementární teorie čísel**, <http://is.muni.cz/el/1431/podzim2012/M6520/um/main-print.pdf>
- Jiří Herman, Radan Kučera, Jaromír Šimša, **Metody řešení matematických úloh**. MU Brno, 2001.
- William Stein, **Elementary Number Theory: Primes, Congruences, and Secrets**, Springer, 2008. Dostupné na <http://wstein.org/ent/ent.pdf>
- Radan Kučera, výukový text k přednášce **Algoritmy teorie čísel**,
<http://www.math.muni.cz/~kucera/texty/ATC10.pdf>

Plán přednášky

- 1 Řád čísla, primitivní kořeny
- 2 Diskrétní logaritmus
- 3 Kvadratické zbytky a nezbytky
- 4 Výpočetní aspekty teorie čísel

Minule

Poslední z této řady tvrzení dává do souvislosti řády dvou čísel a řád jejich součinu:

Lemma

Nechť $m \in \mathbb{N}$, $a, b \in \mathbb{Z}$, $(a, m) = (b, m) = 1$. Jestliže a je řádu r a b je řádu s modulo m , kde $(r, s) = 1$, pak číslo $a \cdot b$ je řádu $r \cdot s$ modulo m .

Minule

Poslední z této řady tvrzení dává do souvislosti řády dvou čísel a řád jejich součinu:

Lemma

Nechť $m \in \mathbb{N}$, $a, b \in \mathbb{Z}$, $(a, m) = (b, m) = 1$. Jestliže a je řádu r a b je řádu s modulo m , kde $(r, s) = 1$, pak číslo $a \cdot b$ je řádu $r \cdot s$ modulo m .

Důkaz.

Označme δ řád čísla $a \cdot b$. Pak $(ab)^\delta \equiv 1 \pmod{m}$ a umocněním obou stran kongruence dostaneme $a^{r\delta} b^{r\delta} \equiv 1 \pmod{m}$. Protože je r řádem čísla a , je $a^r \equiv 1 \pmod{m}$, tj. $b^{r\delta} \equiv 1 \pmod{m}$, a proto $s \mid r\delta$. Z nesoudělnosti r a s plyne $s \mid \delta$. Analogicky dostaneme i $r \mid \delta$, a tedy (opět s využitím nesoudělnosti r, s) $r \cdot s \mid \delta$. Obráceně zřejmě platí $(ab)^{rs} \equiv 1 \pmod{m}$, proto $\delta \mid rs$. Celkem tedy $\delta = rs$.



Minule

Důsledek

Nechť $m \in \mathbb{N}$ a r je nejmenší společný násobek všech řádů modulo m . Pak existuje číslo řádu r modulo m .

Minule

Důsledek

Nechť $m \in \mathbb{N}$ a r je nejmenší společný násobek všech řádů modulo m . Pak existuje číslo řádu r modulo m .

Důkaz.

Stačí pro a řádu s , b řádu t najít prvek řádu $[s, t]$. Nechť $d = (s, t)$, pak tímto prvkem je $a^d \cdot b$. □

Minule

Důsledek

Nechť $m \in \mathbb{N}$ a r je nejmenší společný násobek všech řádů modulo m . Pak existuje číslo řádu r modulo m .

Důkaz.

Stačí pro a řádu s , b řádu t najít prvek řádu $[s, t]$. Nechť $d = (s, t)$, pak tímto prvkem je $a^d \cdot b$. □

Pak všechna $(a, m) = 1$ splňují $a^r \equiv 1 \pmod{m}$, tj. jsou to řešení kongruence

$$x^r \equiv 1 \pmod{m}$$

Zejména nás budou zajímat tzv. primitivní kořeny, tj. čísla mající řád přesně $\varphi(m)$ – to je přesně počet řešení této rovnice.

Primitivní kořeny modulo součin

Příklad

Nechť $m = 35$ a necht' $(a, m) = 1$. Pak podle Eulerovy věty

$$a^4 \equiv 1 \pmod{5}, \quad a^6 \equiv 1 \pmod{7}$$

$$a^{12} \equiv 1 \pmod{5}, \quad a^{12} \equiv 1 \pmod{7} \quad \Rightarrow \quad a^{12} \equiv 1 \pmod{35}.$$

Je tedy každé číslo řádu 12 (případně menšího, ale to vyloučíme časem).

Primitivní kořeny modulo součin

Příklad

Nechť $m = 35$ a nechť $(a, m) = 1$. Pak podle Eulerovy věty

$$a^4 \equiv 1 \pmod{5}, \quad a^6 \equiv 1 \pmod{7}$$

$$a^{12} \equiv 1 \pmod{5}, \quad a^{12} \equiv 1 \pmod{7} \quad \Rightarrow \quad a^{12} \equiv 1 \pmod{35}.$$

Je tedy každé číslo řádu 12 (případně menšího, ale to vyloučíme časem).

Tedy kongruence $x^{12} \equiv 1 \pmod{35}$ stupně 12 má $\varphi(35) = 4 \cdot 6 = 24$ řešení.

Primitivní kořeny modulo součin

Příklad

Nechť $m = 35$ a necht' $(a, m) = 1$. Pak podle Eulerovy věty

$$a^4 \equiv 1 \pmod{5}, \quad a^6 \equiv 1 \pmod{7}$$

$$a^{12} \equiv 1 \pmod{5}, \quad a^{12} \equiv 1 \pmod{7} \quad \Rightarrow \quad a^{12} \equiv 1 \pmod{35}.$$

Je tedy každé číslo řádu 12 (případně menšího, ale to vyloučíme časem).

Tedy kongruence $x^{12} \equiv 1 \pmod{35}$ stupně 12 má $\varphi(35) = 4 \cdot 6 = 24$ řešení.

Věta

Pokud je m dělitelné aspoň dvěma lichými prvočísly, primitivní kořen modulo m neexistuje.

Polynomiální kongruence modulo prvočíslo

Uvažme $f(x) \equiv 0 \pmod{p}$ a vydělme $f(x)$ se zbytkem kořenovým činitelem $(x - a)$:

$$f(x) = (x - a) \cdot g(x) + r \quad \Rightarrow \quad r = f(a)$$

Polynomiální kongruence modulo prvočíslo

Uvažme $f(x) \equiv 0 \pmod{p}$ a vydělme $f(x)$ se zbytkem kořenovým činitelem $(x - a)$:

$$f(x) = (x - a) \cdot g(x) + r \quad \Rightarrow \quad r = f(a)$$

Pokud je a kořenem kongruence, dostaneme

$$f(x) \equiv (x - a) \cdot g(x) \pmod{p}.$$

Protože je p prvočíslo, jsou kořeny $f(x)$ právě a a kořeny $g(x)$, který je stupně o jedna menšího. Protože konstantní polynomy nemají kořeny, má $f(x)$ maximálně tolik kořenů, kolik je jeho stupeň (bacha na $f(x) \equiv 0$).

Polynomiální kongruence modulo prvočíslo

Uvažme $f(x) \equiv 0 \pmod{p}$ a vydělme $f(x)$ se zbytkem kořenovým činitelem $(x - a)$:

$$f(x) = (x - a) \cdot g(x) + r \quad \Rightarrow \quad r = f(a)$$

Pokud je a kořenem kongruence, dostaneme

$$f(x) \equiv (x - a) \cdot g(x) \pmod{p}.$$

Protože je p prvočíslo, jsou kořeny $f(x)$ právě a a kořeny $g(x)$, který je stupně o jedna menšího. Protože konstantní polynomy nemají kořeny, má $f(x)$ maximálně tolik kořenů, kolik je jeho stupeň (bacha na $f(x) \equiv 0$).

Důsledek

$x^2 \equiv 1 \pmod{p}$ má kořeny právě ± 1 .

Věta

Nechť $p \in \mathbb{N}$ je prvočíslo. Pak existuje primitivní kořen modulo p .

Věta

Nechť $p \in \mathbb{N}$ je prvočíslo. Pak existuje primitivní kořen modulo p .

Důkaz.

Nechť r je maximální řád, podle Eulerovy věty $r \mid p - 1$. Pak všech $p - 1$ nenulových zbytkových tříd jsou kořeny

$$x^r \equiv 1 \pmod{p}$$

a podle předchozího $p - 1 \leq r$. □

Věta

Nechť $p \in \mathbb{N}$ je prvočíslo. Pak existuje primitivní kořen modulo p^k .

Věta

Nechť $p \in \mathbb{N}$ je prvočíslo. Pak existuje primitivní kořen modulo p^k .

Důkaz.

Platí $\varphi(p^k) = p^{k-1} \cdot (p - 1)$ a tyto činitele jsou nesoudělné.

Věta

Nechť $p \in \mathbb{N}$ je prvočíslo. Pak existuje primitivní kořen modulo p^k .

Důkaz.

Platí $\varphi(p^k) = p^{k-1} \cdot (p - 1)$ a tyto činitele jsou nesoudělné. Nechť $g \pmod{p}$ je primitivní kořen, tj. prvek řádu $p - 1$. Pak řád $g \pmod{p^k}$ bude násobkem $p - 1$.

Věta

Nechť $p \in \mathbb{N}$ je prvočíslo. Pak existuje primitivní kořen modulo p^k .

Důkaz.

Platí $\varphi(p^k) = p^{k-1} \cdot (p - 1)$ a tyto činitele jsou nesoudělné. Nechť $g \pmod{p}$ je primitivní kořen, tj. prvek řádu $p - 1$. Pak řád $g \pmod{p^k}$ bude násobkem $p - 1$. Stačí najít prvek řádu p^{k-1} . Ukážeme, že je jím $1 + p$; indukcí vzhledem ke $k = 1, 2, \dots$; konkrétně ukážeme:

$$(1 + p)^{p^{k-2}} \equiv 1 + p^{k-1} \not\equiv 1 \pmod{p^k}$$

Věta

Nechť $p \in \mathbb{N}$ je prvočíslo. Pak existuje primitivní kořen modulo p^k .

Důkaz.

Platí $\varphi(p^k) = p^{k-1} \cdot (p - 1)$ a tyto činitele jsou nesoudělné. Nechť $g \pmod{p}$ je primitivní kořen, tj. prvek řádu $p - 1$. Pak řád $g \pmod{p^k}$ bude násobkem $p - 1$. Stačí najít prvek řádu p^{k-1} . Ukážeme, že je jím $1 + p$; indukcí vzhledem ke $k = 1, 2, \dots$; konkrétně ukážeme:

$$(1 + p)^{p^{k-2}} \equiv 1 + p^{k-1} \not\equiv 1 \pmod{p^k}$$

(instance pro $k + 1$ dá $(1 + p)^{p^{k-1}} \equiv 1 + p^k \pmod{p^{k+1}}$).

Věta

Nechť $p \in \mathbb{N}$ je prvočíslo. Pak existuje primitivní kořen modulo p^k .

Důkaz.

Platí $\varphi(p^k) = p^{k-1} \cdot (p - 1)$ a tyto činitele jsou nesoudělné. Nechť $g \pmod{p}$ je primitivní kořen, tj. prvek řádu $p - 1$. Pak řád $g \pmod{p^k}$ bude násobkem $p - 1$. Stačí najít prvek řádu p^{k-1} . Ukážeme, že je jím $1 + p$; indukcí vzhledem ke $k = 1, 2, \dots$; konkrétně ukážeme:

$$(1 + p)^{p^{k-2}} \equiv 1 + p^{k-1} \not\equiv 1 \pmod{p^k}$$

(instance pro $k + 1$ dá $(1 + p)^{p^{k-1}} \equiv 1 + p^k \pmod{p^k}$).

Věta

Nechť $p \in \mathbb{N}$ je prvočíslo. Pak existuje primitivní kořen modulo p^k .

Důkaz.

Platí $\varphi(p^k) = p^{k-1} \cdot (p - 1)$ a tyto činitele jsou nesoudělní. Nechť $g \pmod{p}$ je primitivní kořen, tj. prvek řádu $p - 1$. Pak řád $g \pmod{p^k}$ bude násobkem $p - 1$. Stačí najít prvek řádu p^{k-1} . Ukážeme, že je jím $1 + p$; indukcí vzhledem ke $k = 1, 2, \dots$; konkrétně ukážeme:

$$(1 + p)^{p^{k-2}} \equiv 1 + p^{k-1} \not\equiv 1 \pmod{p^k}$$

(instance pro $k + 1$ dá $(1 + p)^{p^{k-1}} \equiv 1 + p^k \equiv 1 \pmod{p^k}$). □

Věta

Nechť $p \in \mathbb{N}$ je prvočíslo. Pak existuje primitivní kořen modulo p^k .

Důkaz.

Platí $\varphi(p^k) = p^{k-1} \cdot (p - 1)$ a tyto činitele jsou nesoudělné. Nechť $g \pmod{p}$ je primitivní kořen, tj. prvek řádu $p - 1$. Pak řád $g \pmod{p^k}$ bude násobkem $p - 1$. Stačí najít prvek řádu p^{k-1} . Ukážeme, že je jím $1 + p$; indukcí vzhledem ke $k = 1, 2, \dots$; konkrétně ukážeme:

$$(1 + p)^{p^{k-2}} \equiv 1 + p^{k-1} \not\equiv 1 \pmod{p^k}$$

(instance pro $k + 1$ dá $(1 + p)^{p^{k-1}} \equiv 1 + p^k \equiv 1 \pmod{p^k}$). □

Lemma

$$a \equiv b \pmod{p^k} \quad \Rightarrow \quad a^p \equiv b^p \pmod{p^{k+1}}.$$

Hledání primitivního kořene

Sofistikovaná metoda se zatím nezná. Zkoušením po nějaké době uspějeme: Kolik je primitivních kořenů modulo prvočíslo?

Hledání primitivního kořene

Sofistikovaná metoda se zatím nezná. Zkoušením po nějaké době uspějeme: Kolik je primitivních kořenů modulo prvočíslo? Jsou to právě $g^a \pmod{p}$, kde $(a, \varphi p) = 1$, tedy jich je $\varphi(\varphi(p))$. Přitom platí

$$p/\varphi(\varphi(p)) \in O(\log \log p),$$

takže pravděpodobnost, že náhodné číslo bude primitivním kořenem je zhruba

$$1/\log \log p$$

Hledání primitivního kořene

Sofistikovaná metoda se zatím nezná. Zkoušením po nějaké době uspějeme: Kolik je primitivních kořenů modulo prvočíslo? Jsou to právě $g^a \pmod{p}$, kde $(a, \varphi p) = 1$, tedy jich je $\varphi(\varphi(p))$. Přitom platí

$$p/\varphi(\varphi(p)) \in O(\log \log p),$$

takže pravděpodobnost, že náhodné číslo bude primitivním kořenem je zhruba

$$1/\log \log p$$

a počet pokusů potřebných k nalezení primitivního kořene s předem danou pravděpodobností je úměrný $\log \log p$, tedy logaritmický vzhledem k délce vstupu

Hledání primitivního kořene

Sofistikovaná metoda se zatím nezná. Zkoušením po nějaké době uspějeme: Kolik je primitivních kořenů modulo prvočíslo? Jsou to právě $g^a \pmod{p}$, kde $(a, \varphi p) = 1$, tedy jich je $\varphi(\varphi(p))$. Přitom platí

$$p/\varphi(\varphi(p)) \in O(\log \log p),$$

takže pravděpodobnost, že náhodné číslo bude primitivním kořenem je zhruba

$$1/\log \log p$$

a počet pokusů potřebných k nalezení primitivního kořene s předem danou pravděpodobností je úměrný $\log \log p$, tedy logaritmický vzhledem k délce vstupu (ověření toho, zda se vskutku jedná o primitivní kořen trvá déle, viz příště).

Plán přednášky

- 1 Řád čísla, primitivní kořeny
- 2 Diskrétní logaritmus
- 3 Kvadratické zbytky a nezbytky
- 4 Výpočetní aspekty teorie čísel

Definice

Nechť $m \in \mathbb{N}$. Celé číslo $g \in \mathbb{Z}$, $(g, m) = 1$ nazveme *primitivním kořenem* modulo m , pokud je jeho řád modulo m roven $\varphi(m)$.

Definice

Nechť $m \in \mathbb{N}$. Celé číslo $g \in \mathbb{Z}$, $(g, m) = 1$ nazveme *primitivním kořenem modulo m* , pokud je jeho řád modulo m roven $\varphi(m)$.

Lemma

Je-li g primitivní kořen modulo m , pak pro každé číslo $a \in \mathbb{Z}$, $(a, m) = 1$ existuje jediné $x_a \in \mathbb{Z}$, $0 \leq x_a < \varphi(m)$ s vlastností $g^{x_a} \equiv a \pmod{m}$.

Definice

Nechť $m \in \mathbb{N}$. Celé číslo $g \in \mathbb{Z}$, $(g, m) = 1$ nazveme *primitivním kořenem modulo m* , pokud je jeho řád modulo m roven $\varphi(m)$.

Lemma

*Je-li g primitivní kořen modulo m , pak pro každé číslo $a \in \mathbb{Z}$, $(a, m) = 1$ existuje jediné $x_a \in \mathbb{Z}$, $0 \leq x_a < \varphi(m)$ s vlastností $g^{x_a} \equiv a \pmod{m}$. Funkce $a \mapsto x_a$ se nazývá **diskrétní logaritmus**, příp. **index čísla x** (vzhledem k danému m a zafixovanému primitivnímu kořeni g) a je bijekcí mezi množinami $\{a \in \mathbb{Z}; (a, m) = 1, 0 < a < m\}$ a $\{x \in \mathbb{Z}; 0 \leq x < \varphi(m)\}$.*

Definice

Nechť $m \in \mathbb{N}$. Celé číslo $g \in \mathbb{Z}$, $(g, m) = 1$ nazveme *primitivním kořenem* modulo m , pokud je jeho řád modulo m roven $\varphi(m)$.

Lemma

Je-li g primitivní kořen modulo m , pak pro každé číslo $a \in \mathbb{Z}$, $(a, m) = 1$ existuje jediné $x_a \in \mathbb{Z}$, $0 \leq x_a < \varphi(m)$ s vlastností $g^{x_a} \equiv a \pmod{m}$. Funkce $a \mapsto x_a$ se nazývá **diskrétní logaritmus**, příp. *index čísla x (vzhledem k danému m a zafixovanému primitivnímu kořeni g)* a je bijekcí mezi množinami $\{a \in \mathbb{Z}; (a, m) = 1, 0 < a < m\}$ a $\{x \in \mathbb{Z}; 0 \leq x < \varphi(m)\}$.

Důkaz.

Předpokládejme, že pro $x, y \in \mathbb{Z}$, $0 \leq x, y < \varphi(m)$ je $g^x \equiv g^y \pmod{m}$. Z vlastností řádu pak $x \equiv y \pmod{\varphi(m)}$, tj. $x = y$, proto je zobrazení injektivní, a tedy i surjektivní. □

Plán přednášky

- 1 Řád čísla, primitivní kořeny
- 2 Diskrétní logaritmus
- 3 Kvadratické zbytky a nezbytky
- 4 Výpočetní aspekty teorie čísel

Kvadratické kongruence modulo prvočíslo

Věta

Nechť p je liché prvočíslo a $(a, p) = 1$. Kongruence $x^2 \equiv a \pmod{p}$ má řešení, právě když $a^{\frac{p-1}{2}} \equiv 1 \pmod{p}$.

Kvadratické kongruence modulo prvočíslo

Věta

Nechť p je liché prvočíslo a $(a, p) = 1$. Kongruence $x^2 \equiv a \pmod{p}$ má řešení, právě když $a^{\frac{p-1}{2}} \equiv 1 \pmod{p}$.

Důkaz.

Použijeme primitivní kořen g a vyjádříme $x^2 \equiv a \pmod{p}$ pomocí něj: necht' $x \equiv g^\xi$, $a \equiv g^\alpha$, pak kongruence je ekvivalentní

$$(g^\xi)^2 \equiv g^\alpha \pmod{p} \Leftrightarrow 2\xi \equiv \alpha \pmod{p-1}.$$

Protože je $p-1$ sudé, řešení existuje, právě když α je sudé:

$$\alpha \equiv 0 \pmod{2} \Leftrightarrow \frac{p-1}{2} \cdot \alpha \equiv 0 \pmod{p-1}.$$

$$\Leftrightarrow a^{\frac{p-1}{2}} \equiv g^{\frac{p-1}{2} \cdot \alpha} \equiv g^0 \equiv 1 \pmod{p}. \square$$

Legendreův symbol

Věta

Nechť p je liché prvočíslo a $(a, p) = 1$. Kongruence $x^2 \equiv a \pmod{p}$ má řešení, právě když $a^{\frac{p-1}{2}} \equiv 1 \pmod{p}$.

Legendreův symbol

Věta

Nechť p je liché prvočíslo a $(a, p) = 1$. Kongruence $x^2 \equiv a \pmod{p}$ má řešení, právě když $a^{\frac{p-1}{2}} \equiv 1 \pmod{p}$.

Definice

Definujeme $\left(\frac{a}{p}\right) = \begin{cases} 1 & a \text{ kvadratický zbytek modulo } p \\ -1 & a \text{ kvadratický nezbytek modulo } p. \\ 0 & a \text{ soudělné s } p \end{cases}$

Legendreův symbol

Věta

Nechť p je liché prvočíslo a $(a, p) = 1$. Kongruence $x^2 \equiv a \pmod{p}$ má řešení, právě když $a^{\frac{p-1}{2}} \equiv 1 \pmod{p}$.

Definice

Definujeme $\left(\frac{a}{p}\right) = \begin{cases} 1 & a \text{ kvadratický zbytek modulo } p \\ -1 & a \text{ kvadratický nezbytek modulo } p. \\ 0 & a \text{ soudělné s } p \end{cases}$

Jednoduchým důsledkem věty dostáváme $\left(\frac{a}{p}\right) \equiv a^{\frac{p-1}{2}} \pmod{p}$:
protože $(a^{\frac{p-1}{2}})^2 \equiv a^{p-1} \equiv 1$, je $a^{\frac{p-1}{2}}$ rovno ± 1 .

Legendreův symbol

Důsledek

$\left(\frac{-1}{p}\right) = +1$, resp. -1 , pokud $p \equiv 1 \pmod{4}$, resp. $p \equiv 3 \pmod{4}$.
Tedy kongruence $x^2 \equiv -1 \pmod{p}$ má řešení, právě když p dává
po dělení čtyřmi zbytek 1.

Legendreův symbol

Důsledek

$\left(\frac{-1}{p}\right) = +1$, resp. -1 , pokud $p \equiv 1 \pmod{4}$, resp. $p \equiv 3 \pmod{4}$.
Tedy kongruence $x^2 \equiv -1 \pmod{p}$ má řešení, právě když p dává po dělení čtyřmi zbytek 1.

Počítání Legendreova symbolu je jednoduché s následujícími pravidly:

- $\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right)$,
- $a \equiv b \pmod{p} \Rightarrow \left(\frac{a}{p}\right) = \left(\frac{b}{p}\right)$,
- $\left(\frac{2}{p}\right) = (-1)^{\frac{p^2-1}{8}}$,
- $\left(\frac{q}{p}\right) = \left(\frac{p}{q}\right) \cdot (-1)^{\frac{p-1}{2} \frac{q-1}{2}}$.

Jacobiho symbol

Definice

a libovolné, $n = p_1 \cdots p_k$ liché; definujeme

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right) \cdots \left(\frac{a}{p_k}\right).$$

Jacobiho symbol

Definice

a libovolné, $n = p_1 \cdots p_k$ liché; definujeme

$$\left(\frac{a}{n}\right) = \left(\frac{a}{p_1}\right) \cdots \left(\frac{a}{p_k}\right).$$

Počítání Jacobiho symbolu je jednoduché s následujícími pravidly:

- $\left(\frac{ab}{n}\right) = \left(\frac{a}{n}\right)\left(\frac{b}{n}\right)$,
- $a \equiv b \pmod{n} \Rightarrow \left(\frac{a}{n}\right) = \left(\frac{b}{n}\right)$,
- $\left(\frac{2}{n}\right) = (-1)^{\frac{n^2-1}{8}}$,
- $\left(\frac{m}{n}\right) \cdot \left(\frac{n}{m}\right) = (-1)^{\frac{m-1}{2} \frac{n-1}{2}}$ pro m, n lichá.

Jacobiho symbol

Příklad

Spočtěte $\left(\frac{219}{383}\right)$.

Jacobiho symbol

Příklad

Spočtěte $\left(\frac{219}{383}\right)$.

Příklad

Spočtěte $\left(\frac{3}{p}\right)$ pro prvočíslo p , tj. rozhodněte, kdy je 3 kvadratický zbytek modulo prvočíslo p .

Testování prvočíselnosti

test

Je-li p prvočíslo, pak $a^{p-1} \equiv 1 \pmod{p}$ (pro a náhodné nesoudělné s p).

Testování prvočíselnosti

test

Je-li p prvočíslo, pak $a^{p-1} \equiv 1 \pmod{p}$ (pro a náhodné nesoudělné s p).

složitější test

Je-li p prvočíslo, pak $a^{\frac{p-1}{2}} \equiv \pm 1 \pmod{p}$ (pro a náhodné nesoudělné s p).

Testování prvočíselnosti

test

Je-li p prvočíslo, pak $a^{p-1} \equiv 1 \pmod{p}$ (pro a náhodné nesoudělné s p).

složitější test

Je-li p prvočíslo, pak $a^{\frac{p-1}{2}} \equiv \pm 1 \pmod{p}$ (pro a náhodné nesoudělné s p).

ještě složitější test

Je-li p prvočíslo, pak $a^{\frac{p-1}{2}} \equiv \left(\frac{a}{p}\right) \pmod{p}$ (pro a náhodné nesoudělné s p).

Plán přednášky

- 1 Řád čísla, primitivní kořeny
- 2 Diskrétní logaritmus
- 3 Kvadratické zbytky a nezbytky
- 4 Výpočetní aspekty teorie čísel**

Základní úlohy výpočetní teorie čísel

V mnoha praktických úlohách využívajících výsledky teorie čísel je zapotřebí umět rychle provést jeden či více z následujících výpočtů:

- 1 běžné aritmetické operace (součet, součin, dělení se zbytkem) na celých číslech,

Základní úlohy výpočetní teorie čísel

V mnoha praktických úlohách využívajících výsledky teorie čísel je zapotřebí umět rychle provést jeden či více z následujících výpočtů:

- 1 běžné aritmetické operace (součet, součin, dělení se zbytkem) na celých číslech,
- 2 zbytek mocniny celého čísla a na přirozené číslo n po dělení daným m .

Základní úlohy výpočetní teorie čísel

V mnoha praktických úlohách využívajících výsledky teorie čísel je zapotřebí umět rychle provést jeden či více z následujících výpočtů:

- 1 běžné aritmetické operace (součet, součin, dělení se zbytkem) na celých číslech,
- 2 zbytek mocniny celého čísla a na přirozené číslo n po dělení daným m .
- 3 inverzi celého čísla a modulo $m \in \mathbb{N}$,

Základní úlohy výpočetní teorie čísel

V mnoha praktických úlohách využívajících výsledky teorie čísel je zapotřebí umět rychle provést jeden či více z následujících výpočtů:

- 1 běžné aritmetické operace (součet, součin, dělení se zbytkem) na celých číslech,
- 2 zbytek mocniny celého čísla a na přirozené číslo n po dělení daným m .
- 3 inverzi celého čísla a modulo $m \in \mathbb{N}$,
- 4 největší společný dělitel dvou celých čísel (a případně koeficienty do Bezoutovy rovnosti),

Základní úlohy výpočetní teorie čísel

V mnoha praktických úlohách využívajících výsledky teorie čísel je zapotřebí umět rychle provést jeden či více z následujících výpočtů:

- 1 běžné aritmetické operace (součet, součin, dělení se zbytkem) na celých číslech,
- 2 zbytek mocniny celého čísla a na přirozené číslo n po dělení daným m .
- 3 inverzi celého čísla a modulo $m \in \mathbb{N}$,
- 4 největší společný dělitel dvou celých čísel (a případně koeficienty do Bezoutovy rovnosti),
- 5 rozhodnout o daném čísle, je-li prvočíslo nebo složené,

Základní úlohy výpočetní teorie čísel

V mnoha praktických úlohách využívajících výsledky teorie čísel je zapotřebí umět rychle provést jeden či více z následujících výpočtů:

- 1 běžné aritmetické operace (součet, součin, dělení se zbytkem) na celých číslech,
- 2 zbytek mocniny celého čísla a na přirozené číslo n po dělení daným m .
- 3 inverzi celého čísla a modulo $m \in \mathbb{N}$,
- 4 největší společný dělitel dvou celých čísel (a případně koeficienty do Bezoutovy rovnosti),
- 5 rozhodnout o daném čísle, je-li prvočíslo nebo složené,
- 6 v případě složenosti rozložit dané číslo na součin prvočísel.

Základní aritmetické operace

Základní aritmetické operace se i na velkých číslech obvykle provádějí obdobně jako jsme se to učili na základní a střední škole, kdy umíme sčítat v *lineárním*, násobit a dělit se zbytkem v *kvadratickém* čase.

Základní aritmetické operace

Základní aritmetické operace se i na velkých číslech obvykle provádějí obdobně jako jsme se to učili na základní a střední škole, kdy umíme sčítat v *lineárním*, násobit a dělit se zbytkem v *kvadratickém* čase. Pro **násobení**, které je základem mnoha dalších operací, existují asymptoticky rychlejší algoritmy (typu *rozděl a panuj*) - např. první takový Karatsubův (1960) časové náročnosti $\Theta(n^{\log_2 3})$

Základní aritmetické operace

Základní aritmetické operace se i na velkých číslech obvykle provádějí obdobně jako jsme se to učili na základní a střední škole, kdy umíme sčítat v *lineárním*, násobit a dělit se zbytkem v *kvadratickém* čase. Pro **násobení**, které je základem mnoha dalších operací, existují asymptoticky rychlejší algoritmy (typu *rozděl a panuj*) - např. první takový Karatsubův (1960) časové náročnosti $\Theta(n^{\log_2 3})$ nebo algoritmus Schönhage-Strassenův (1971) časové náročnosti $\Theta(n \log n \log \log n)$, který využívá tzv. Fast Fourier Transform. Ten je ale přes svou asymptotickou převahu výhodný až pro násobení čísel majících alespoň desítky tisíc cifer (a používá se tak např. v GIMPS).

Základní aritmetické operace

Základní aritmetické operace se i na velkých číslech obvykle provádějí obdobně jako jsme se to učili na základní a střední škole, kdy umíme sčítat v *lineárním*, násobit a dělit se zbytkem v *kvadratickém* čase. Pro **násobení**, které je základem mnoha dalších operací, existují asymptoticky rychlejší algoritmy (typu *rozděl a panuj*) - např. první takový Karatsubův (1960) časové náročnosti $\Theta(n^{\log_2 3})$ nebo algoritmus Schönhage-Strassenův (1971) časové náročnosti $\Theta(n \log n \log \log n)$, který využívá tzv. Fast Fourier Transform. Ten je ale přes svou asymptotickou převahu výhodný až pro násobení čísel majících alespoň desítky tisíc cifer (a používá se tak např. v GIMPS). Pěkný přehled je např. na http://en.wikipedia.org/wiki/Computational_complexity_of_mathematical_operations

GCD a modulární inverze

Jak už jsme ukazovali dříve, výpočet řešení kongruence $a \cdot x \equiv 1 \pmod{m}$ s neznámou x lze snadno (díky Bezoutově větě) převést na výpočet největšího společného dělitele čísel a a m a na hledání koeficientů k, l do Bezoutovy rovnosti $k \cdot a + l \cdot m = 1$ (nalezené k je pak onou hledanou inverzí a modulo m).

GCD a modulární inverze

Jak už jsme ukazovali dříve, výpočet řešení kongruence $a \cdot x \equiv 1 \pmod{m}$ s neznámou x lze snadno (díky Bezoutově větě) převést na výpočet největšího společného dělitele čísel a a m a na hledání koeficientů k, l do Bezoutovy rovnosti $k \cdot a + l \cdot m = 1$ (nalezené k je pak onou hledanou inverzí a modulo m).

```
function extended_gcd(a, m)
  if m == 0
    return (1, 0)
  else
    (q, r) := divide(a, m)
    (k, l) := extended_gcd(m, r)
    return (l, k - q * l)
```

Podrobná analýza (viz např. [Knuth] nebo [Wiki]) ukazuje, že tento algoritmus je **kvadratické** časové složitosti.

Modulární umocňování

Modulární umocňování je, jak jsme již viděli dříve, velmi využívaná operace mj. při ověřování, zda je dané číslo prvočíslo nebo číslo složené. Jedním z efektivních algoritmů je tzv. **modulární umocňování zprava doleva**:

Modulární umocňování

Modulární umocňování je, jak jsme již viděli dříve, velmi využívaná operace mj. při ověřování, zda je dané číslo prvočíslo nebo číslo složené. Jedním z efektivních algoritmů je tzv. **modulární umocňování zprava doleva**:

```
function modular_pow(base, exponent, modulus)
    result := 1
    while exponent > 0
        if (exponent mod 2 == 1):
            result := (result * base) mod modulus
        exponent := exponent >> 1
        base = (base * base) mod modulus
    return result
```

Algoritmus modulárního umocňování je založen na myšlence, že např. při počítání $2^{64} \pmod{1000}$

- není třeba nejprve počítat 2^{64} a poté jej vydělit se zbytkem číslem 1000, ale lépe je postupně násobit „dvojky“ a kdykoliv je výsledek větší než 1000, provést redukci modulo 1000,

Algoritmus modulárního umocňování je založen na myšlence, že např. při počítání $2^{64} \pmod{1000}$

- není třeba nejprve počítat 2^{64} a poté jej vydělit se zbytkem číslem 1000, ale lépe je postupně násobit „dvojky“ a kdykoliv je výsledek větší než 1000, provést redukci modulo 1000,
- ale zejména, že není třeba provádět takové množství násobení (v tomto případě 63 naivních násobení je možné nahradit pouze šesti umocněními na druhou, neboť

$$2^{64} = ((((((2^2)^2)^2)^2)^2)^2)^2.$$

Příklad (Ukázka průběhu algoritmu)

Vypočtěme $2^{560} \pmod{561}$.

Příklad (Ukázka průběhu algoritmu)

VypočtĚme $2^{560} \pmod{561}$. Protože $560 = (1000110000)_2$, dostaneme uvedeným algoritmem

exponent	base	result	exp's last digit
560	2	1	0
280	4	1	0
140	16	1	0
70	256	1	0
35	460	1	1
17	103	460	1
8	511	256	0
4	256	256	0
2	460	256	0
1	103	256	1
0	511	1	0

Příklad (Ukázka průběhu algoritmu)

Vypočtěme $2^{560} \pmod{561}$. Protože $560 = (1000110000)_2$, dostaneme uvedeným algoritmem

exponent	base	result	exp's last digit
560	2	1	0
280	4	1	0
140	16	1	0
70	256	1	0
35	460	1	1
17	103	460	1
8	511	256	0
4	256	256	0
2	460	256	0
1	103	256	1
0	511	1	0

A tedy $2^{560} \equiv 1 \pmod{561}$.

Efektivita modulárního umocňování

V průběhu algoritmu se pro každou binární číslici exponentu provede umocnění základu na druhou modulo n (což je operace proveditelná v nejhůře kvadratickém čase), a pro každou „jedničku“ v binárním zápisu navíc provede jedno násobení. Celkově jsme tedy schopni provést modulární umocňování nejhůře v **kubickém** čase.