Počítání modulo n a jeho časová složitost

3. a 4. přednáška z kryptografie

Obsah

Umocňování v $\mathbb{Z}_n$

Při sčítání a násobení v $\mathbb{Z}_n$ můžeme čísla nahradit jejich zbytky modulo n. Můžeme nějak zmenšit exponent při umocňování?

Čísel v $\mathbb{Z}_n$ je konečně mnoho, výsledky mocnin se musí opakovat:

$$\text{Existují} \quad k > l \in \mathbb{N} \quad \text{tak, že} \quad a^k = a^l.$$

Pokud je a invertibilní v $\mathbb{Z}_n$, získáme odtud $a^{k-l} = 1$. Mocniny čísla a se cyklí s periodou $k - l$.
Jaká je společná perioda pro všechna invertibilní $a \in \mathbb{Z}_n$?

Euler-Fermatova věta

Malá Fermatova věta

Nechť p je prvočíslo, $k \in \mathbb{Z}$.
Pro každé $a \neq kp$ je $a^{p-1} \equiv 1 \pmod{p}$.

Euler-Fermatova věta

Pro každé $a \in \mathbb{Z}_n$, a nesoudělné s n, je $a^{\varphi(n)} = 1$ v $\mathbb{Z}_n$.

Aneb: Je-li základ nesoudělný s n, můžeme exponent zmenšit modulo $\varphi(n)$.

Euler-Fermatova věta

Eulerova funkce

$\varphi : \mathbb{N} \to \mathbb{N} : \varphi(n) =$ počet čísel mezi 0 až (n-1) nesoudělných s n

Pro výpočet Eulerovy funkce platí následující vzorce:

- $\varphi(p) = p - 1$ pro p prvočíslo
- $\varphi(p^k) = p^k - p^{k-1} = p^{k-1}(p-1)$ pro p prvočíslo a $k \in \mathbb{N}$
- $\varphi(n \cdot m) = \varphi(n) \cdot \varphi(m)$ pro $n, m \in \mathbb{N}$ navzájem nesoudělná

Příklady

1) $\varphi(100) = \varphi(2^2 \cdot 5^2) = (4-2) \cdot (25-5) = 40$; $\varphi(1) = 1$.
Známe-li prvočíselný rozklad čísla n, umíme spočítat $\varphi(n)$.
2) $5^{64} = 5^4 = 13$ v $\mathbb{Z}_{18}$, protože $\gcd(5, 18) = 1$ a $\varphi(18) = 6$.

Euler-Fermatova věta

Eulerova věta

Nechť $(G, \circ)$ je konečná grupa o n prvcích s neutrálním prvkem 1.
Pro každé $a \in G$ platí: $a^n = \underbrace{a \circ a \circ \ldots \circ a}_{n\text{-krát}} = 1$ v G.

Euler-Fermatova věta je speciálním případem Eulerovy věty aplikované na grupu $(\mathbb{Z}_n^*, \cdot)$ invertibilních prvků v monoidu $(\mathbb{Z}_n, \cdot)$.

$$\mathbb{Z}_n^* = \{a \in \mathbb{Z}_n; a \text{ je nesoudělné s } n\}$$

Počet prvků této grupy $|\mathbb{Z}_n^*| = \varphi(n)$, neutrální prvek je 1.

Hledání inverzních prvků v $\mathbb{Z}_n$

Poznámka

Euler-Fermatovu větu lze použít pro počítání inverzních prvků v $\mathbb{Z}_n$.

Je-li a nesoudělné s n, pak $a^{-1} = a^{\varphi(n)-1}$ v $\mathbb{Z}_n$.

K výpočtu lze použít algoritmus opakovaných čtverců.

Poznámka

Pokud a není invertibilní v $\mathbb{Z}_n$, pak také existují exponenty $k > l$ tak, že $a^k = a^l$. Mocniny prvku a se cyklí s periodou $k - l$, ale žádná mocnina není rovna 1, tj. $a^k \neq 1$ v $\mathbb{Z}_n$ pro každé $k > 0$.
Jinak, kdyby $a^k = 1$ v $\mathbb{Z}_n$, tak by existovalo $a^{-1} = a^{k-1}$.

Algoritmus opakovaných čtverců

Počítáme a^b v $\mathbb{Z}_n$ postupným umocňováním na druhou.

Napíšeme exponent binárně: $b = (b_{k-1} \ldots b_0)_2$
Vytvoříme posloupnost příkazů X="times a in $\mathbb{Z}_n$",
S="square in $\mathbb{Z}_n$" takto:
Do každé mezery v binárním zápisu dáme příkaz S, tím vznikne k přihrádek. Do přihrádky dáme příkaz X, právě když je na příslušném místě binárního zápisu 1, jinak necháme přihrádku prázdnou.
Začneme od $a^0 = 1$ a vykonáváme příkazy po řadě zleva doprava.

Příklad

Číslu $b = 13 = (1101)_2$ odpovídá posloupnost $XSXSSX$.
2^{13} v $\mathbb{Z}_{20}$: $1 \xrightarrow{X} 2 \xrightarrow{S} 4 \xrightarrow{X} 8 \xrightarrow{S} 4 \xrightarrow{S} 16 \xrightarrow{X} 12 = 2^{13}$.

Algoritmus opakovaných čtverců

Vstup: přirozená čísla a, b, n
Výstup: a^b v $\mathbb{Z}_n$
Nechť $b = (b_{k-1} \dots b_0)_2$ je binární rozvoj exponentu b.

- $c \leftarrow 1$
- for $i \leftarrow k-1$ down to 0 do
  - $c \leftarrow c^2$ in $\mathbb{Z}_n$
  - if $b_i = 1$ then $c \leftarrow ca$ in $\mathbb{Z}_n$
- output c

Časová složitost - provádí se nejvýše $2\log_2(b)$ násobení v $\mathbb{Z}_n$.
Prostorová složitost - počítá se s čísly menšími než n^2.

Čínská věta o zbytcích

Čínská věta o zbytcích

Nechť $n_1, \dots, n_k$ jsou po dvou nesoudělná přirozená čísla, $a_1, \dots, a_k$ jsou libovolná přirozená čísla.
Pak soustava rovnic

$$x \equiv a_i \pmod{n_i} \quad \text{pro všechna} \quad 1 \leq i \leq k$$

má řešení.
Navíc každá dvě řešení a, b jsou kongruentní modulo $n = \prod_{i=1}^{k} n_i$.

Čínská věta o zbytcích

Důkaz

Důkaz existence řešení nám dává universální návod na řešení zbytkových soustav, proto ho zde uvedeme.

Nejprve pro každé $1 \leq i \leq k$ vyřešíme speciální zbytkovou soustavu:
$x \equiv 1 \pmod{n_i}$ a $x \equiv 0 \pmod{n_j}$ pro $j \neq i$.
Řešení $i-$té zbytkové soustavy, označme ho jako q_i, nalezneme takto:
$q_i = (\prod_{j \neq i} n_i) t_i$, kde $t_i = (\prod_{j \neq i} n_i)^{-1}$ v $\mathbb{Z}_{n_i}$
(díky nesoudělnosti čísel $n_1, \dots, n_k$ tento inverzní prvek existuje).

Nyní se snadno nahlédne, že $a = \sum_{i=1}^{k} a_i q_i$ řeší zadanou zbytkovou soustavu.

Čínská věta o zbytcích

Příklad

Řešte zbytkovou soustavu:
$x \equiv 2 \pmod{4}$, $x \equiv 0 \pmod{5}$, $x \equiv 1 \pmod{9}$, $x \equiv 2 \pmod{11}$

Místo q_i použijeme značení q_{n_i}.
$q_4 = 5 \cdot 9 \cdot 11 \cdot t$, kde t dopočteme v $\mathbb{Z}_4$: $t = (1 \cdot 1 \cdot 3)^{-1} = 3$.
Analogicky spočteme ostatní q_{n_i}.
Vyjde $q_4 = 1485$, $q_5 = 396$, $q_9 = 1540$, $q_{11} = 540$.

Potom $x = 2q_4 + 0q_5 + 1q_9 + 2q_{11} = 1630$ je jediné řešení v $\mathbb{Z}_{1980}$, protože $1980 = 4 \cdot 5 \cdot 9 \cdot 11$.

Zbytkové soustavy obecně

Pokud $n_1, \ldots, n_k$ nejsou nutně po dvou nesoudělná přirozená čísla, pak soustava rovnic

$$x \equiv a_i \pmod{n_i} \quad \text{pro všechna} \quad 1 \leq i \leq k$$

může, ale nemusí mít řešení. Má-li soustava řešení, pak každá dvě řešení a, b jsou kongruentní modulo $n = \operatorname{lcm}(n_1, \ldots, n_k)$.

Příklady

1) Soustava $x \equiv 1 \pmod 2$, $x \equiv 0 \pmod 4$ jistě nemá řešení.

2) Soustava $x \equiv 1 \pmod 2$, $x \equiv 3 \pmod 4$, $x \equiv 1 \pmod 5$ má řešení $x = 11 + 20t$ pro každé $t \in \mathbb{Z}$.
Nalezneme ho vyřešením dvou Diofantických rovnic, které získáme ze vztahů: $x = 2k + 1 = 4l + 3 = 5m + 1$ pro $k, l, m \in \mathbb{Z}$

Reziduální aritmetika

Tvrzení

Nechť $n_1, \ldots, n_k$ jsou po dvou nesoudělná přirozená čísla a nechť $n = \prod_{i=1}^{k} n_i$. Definujme zobrazení:

$$\theta : \mathbb{Z}_n \to \mathbb{Z}_{n_1} \times \ldots \times \mathbb{Z}_{n_k} : [a]_n \mapsto ([a]_{n_1}, \ldots, [a]_{n_k})$$

1. Definice je korektní, nezávisí na volbě representanta třídy $[a]_n$.
2. Zobrazení θ je bijekce (vzájemně jednoznačné).
3. Pro všechna $\alpha, \beta \in \mathbb{Z}_n$, kde $\theta(\alpha) = (\alpha_1, \ldots, \alpha_k)$, $\theta(\beta) = (\beta_1, \ldots, \beta_k)$, platí:
   $\theta(\alpha + \beta) = (\alpha_1 + \beta_1, \ldots, \alpha_k + \beta_k)$, $\theta(0) = (0, \ldots, 0)$,
   $\theta(-\alpha) = (-\alpha_1, \ldots, -\alpha_k)$;
   $\theta(\alpha \cdot \beta) = (\alpha_1 \cdot \beta_1, \ldots, \alpha_k \cdot \beta_k)$, $\theta(1) = (1, \ldots, 1)$,
   $\alpha \in \mathbb{Z}_n^*$ iff každé $\alpha_i \in \mathbb{Z}_{n_i}^*$. Tehdy $\theta(\alpha^{-1}) = (\alpha_1^{-1}, \ldots, \alpha_k^{-1})$.

Reziduální aritmetika

Poznámky

- Zobrazení θ bychom mohli definovat přes základní representanty tříd v $\mathbb{Z}_n$, tj. zbytky po dělení n, takto:
  Nechť $n_1, \ldots, n_k$ jsou po dvou nesoudělná přirozená čísla, nechť $n = \prod_{i=1}^{k} n_i$.
  Pro libovolné $0 \leq a < n$ označme jeho zbytek po dělení číslem n_i jako a_i, tedy $a \equiv a_i \pmod{n_i}$, $0 \leq a_i < n_i$. Pak zobrazení
  $$\theta : \mathbb{Z}_n \to \mathbb{Z}_{n_1} \times \ldots \times \mathbb{Z}_{n_k} : a \mapsto (a_1, \ldots, a_k)$$
  je tzv. *(Čínské) zbytkové zobrazení*.
- Předchozí tvrzení říká, že zbytkové zobrazení θ je okruhový izomorfismus, který respektuje invertibilní prvky.

Reziduální aritmetika

Poznámky

- Restrikce zobrazení θ na množinu $\mathbb{Z}_n^*$ je bijekcí množiny $\mathbb{Z}_n^*$ na množinu $\mathbb{Z}_{n_1}^* \times \ldots \times \mathbb{Z}_{n_k}^*$, které je grupovým izomorfismem.
- Jsou-li n, m navzájem nesoudělná přirozená čísla, pak $\varphi(n \cdot m) = \varphi(n) \cdot \varphi(m)$.

Důsledek

Nechť $n = \prod_{i=1}^{k} p_i^{e_i}$, kde p_i jsou navzájem různá prvočísla, aneb jejich mocniny jsou po dvou nesoudělné.
Chceme-li počítat v $\mathbb{Z}_n$, stačí umět počítat v příslušných $\mathbb{Z}_{p_i^{e_i}}$ a použít Čínský zbytkový izomorfismus.

Reziduální aritmetika

Reziduální aritmetika

Chceme počítat s čísly v rozmezí $-M \leq c < M$, respektive s čísly v rozmezí $0 \leq c < 2M$.
Zvolíme sadu po dvou nesoudělných čísel $n_1, \ldots, n_k$ tak, aby $n = \prod_{i=1}^{k} n_i > 2M$ (jen o trochu větší). Spočteme pro tuto sadu universální koeficienty q_i, $1 \leq i \leq k$.
Veškeré výpočty provádíme reziduálně v každém $\mathbb{Z}_{n_i}$ a výsledek v $\mathbb{Z}_n$ dopočteme pomocí Čínské věty o zbytcích.
Víme-li, že výsledky jsou v rozmezí $-M$ až M, pak číslo $M \leq c < 2M$ odpovídá výsledku $-M \leq c - n < 0$.

Reziduální aritmetika

Příklad

V $\mathbb{Z}_{1980}$ spočtěte $a \cdot b$, a^b, a^{-1} pro čísla
$a = 31313131313$, $b = 123456789$.

Víme, že $1980 = 4 \cdot 5 \cdot 9 \cdot 11$, tedy $\mathbb{Z}_{1980} \cong \mathbb{Z}_4 \times \mathbb{Z}_5 \times \mathbb{Z}_9 \times \mathbb{Z}_{11}$.
Universální q_{n_i} pro tuto nesoudělnou sadu čísel jsou
$q_4 = 1485$, $q_5 = 396$, $q_9 = 1540$, $q_{11} = 540$.

$\theta(a) = (1, 3, 5, 2)$, $\theta(b) = (1, 4, 0, 5)$.

$\theta(a \cdot b) = (1 \cdot 1, 3 \cdot 4, 5 \cdot 0, 2 \cdot 5) = (1, 2, 0, -1)$.
Odtud $a \cdot b = 1q_4 + 2q_5 + 0q_9 - 1q_{11} = 1737$ v $\mathbb{Z}_{1980}$.

$\theta(a^b) = (1^b, 3^b, 5^b, 2^b) = (1^1, 3^1, 5^3, 2^9) = (1, 3, -1, 6)$, použili jsme Euler-Fermatovu větu v každém $\mathbb{Z}_{n_i}$. Odtud $a^b = 413$ v $\mathbb{Z}_{1980}$.

$\theta(a^{-1}) = (1^{-1}, 3^{-1}, 5^{-1}, 2^{-1}) = (1, 2, 2, 6)$, $a^{-1} = 677$ v $\mathbb{Z}_{1980}$.

Reziduální aritmetika

Věty o dělitelnosti

Nechť $a = \sum_{i=0}^{k} a_i \cdot 10^i$, kde a_i jsou číslice, cifry.

- $a \equiv \sum_{i=0}^{k} a_i \pmod 3$, *resp.* $\pmod 9$
- $a \equiv \sum_{i=0}^{k} (-1)^i a_i \pmod{11}$

Nechť $a = \sum_{i=0}^{k} t_i \cdot 1000^i$, kde t_i jsou trojčíslí, trojcifří.

- $a \equiv \sum_{i=0}^{k} (-1)^i t_i \pmod 7$, *resp.* $\pmod{13}$

Asymptotická notace

Definice

Nechť f a g jsou reálné funkce a $g(x) \geq 0$ (stačí, aby funkce byly definované a g nezáporná "pro všechna dostatečně velká x").

- $f \in O(g)$, když existuje $c > 0$ a existuje $x_0 \in \mathbb{R}$ tak,že pro všechna $x \geq x_0$ je $|f(x)| \leq cg(x)$.
- $f \in \Omega(g)$, když existuje $c > 0$ a existuje $x_0 \in \mathbb{R}$ tak,že pro všechna $x \geq x_0$ je $f(x) \geq cg(x)$.
- $f \in \Theta(g)$, když existují $c, d > 0$ a existuje $x_0 \in \mathbb{R}$ tak,že pro všechna $x \geq x_0$ je $dg(x) \leq f(x) \leq cg(x)$.

Asymptotická notace

Definice

Nechť f a g jsou reálné funkce a $g(x) \geq 0$ (stačí, aby funkce byly definované a g nezáporná "pro všechna dostatečně velká x").

- $f \in o(g)$, když pro každou $c > 0$ existuje $x_0 \in \mathbb{R}$ tak, že pro všechna $x \geq x_0$ je $|f(x)| \leq cg(x)$.
- $f \in o(g)$, když $\lim_{x\to\infty} \frac{f(x)}{g(x)} = 0$.
- $f \sim g$ (asymptoticky ekvivalentní), když $\lim_{x\to\infty} \frac{f(x)}{g(x)} = 1$.

Representace čísel

Délka čísla

Délka celého čísla a je počet bitů v binární representaci $|a|$, tedy

- $\mathrm{len}(a) = \lfloor \log_2 |a| \rfloor + 1$, pokud $a \neq 0$
- $\mathrm{len}(a) = 1$, pokud $a = 0$

Representace velkých celých čísel

Velká celá čísla se v paměti uchovávají jako vektor slov délky $\mathrm{len}(B)$ spolu se znaménkovým bitem:

$$a = \pm \sum_{i=0}^{k-1} a_i B^i = \pm(a_{k-1}, \ldots, a_0)_B$$

Např. v jazycích *C* a *Java* na 32−bitových počítačích pro typ *Integer* je $B = 2^{15}$. Potom $\mathrm{len}(a) = k\,\mathrm{len}(B) = O(k)$.

Základní aritmetické operace v $\mathbb{Z}$

Tvrzení

Nechť a, b jsou celá čísla. Předpokládejme, že sečtení či vynásobení dvou bitů trvá 1 jednotku času.

- $a \pm b$ se spočte v čase $O(\mathrm{len}(a) + \mathrm{len}(b))$.
- $a \cdot b$ se spočte v čase $O(\mathrm{len}(a)\,\mathrm{len}(b))$.
- Pokud $b \neq 0$, $a = qb + r$, částečný podíl q a zbytek z se spočtou v čase $O(\mathrm{len}(b)\,\mathrm{len}(q))$.
  Přitom $\mathrm{len}(a) - \mathrm{len}(b) - 1 \leq \mathrm{len}(q) \leq \mathrm{len}(a) - \mathrm{len}(b) + 1$.
- Násobení a dělení čísla a mocninou 2^n se spočte v čase $O(\mathrm{len}(a))$, neboť je to jen posun bitů doleva či doprava.

Základní aritmetické operace v $\mathbb{Z}$

Rychlejší násobení

- Klasický algoritmus pro násobení dvou čísel délky l v čase $O(l^2)$ není nejrychlejší. Pro naše odhady složitosti algoritmů bude postačující (budeme tedy mít horní odhad času).
- Karatsubův algoritmus pro násobení dvou čísel délky l potřebuje čas $O(l^{\log_2(3)})$, přitom $\log_2(3) \doteq 1,58$.
- Při počítání s velkými čísly representovanými v $B = 2^{15}$−ární soustavě se vynásobení dvou slov délky 15 se děje v rámci jednoho 32−bitového slova. Můžeme předpokládat, že trvá jednotku času. Pak bude multiplikativní konstanta v odhadech času $\frac{1}{B}$−krát menší. Volba B neovlivní teoretické výpočty, ale hraje významnou roli v praxi.

Základní aritmetické operace v $\mathbb{Z}_n$

Tvrzení

Nechť a, b jsou čísla ze $\mathbb{Z}_n$ ($0 \leq a, b < n$), exponent $e \in \mathbb{N}$.
Operace provádíme v $\mathbb{Z}_n$ a výsledek je v rozmezí $0 \leq c < n$.

- $a \pm b$ se spočte v čase $O(\mathrm{len}(n))$.
- $a \cdot b$ se spočte v čase $O(\mathrm{len}(n)^2)$.
- a^e se spočte v čase $O(\mathrm{len}(e)\,\mathrm{len}(n)^2)$ algoritmem opakovaných čtverců.
- Je-li $\gcd(a, n) = 1$, pak se a^e spočte v čase $O(\mathrm{len}(e)\,\mathrm{len}(n) + \mathrm{len}(n)^3)$ algoritmem opakovaných čtverců s použitím Euler-Fermatovy věty.
- Je-li $\gcd(a, n) = 1$, pak se a^{-1} spočte v čase $O(\mathrm{len}(n)^3)$ algoritmem opakovaných čtverců.

Časová složitost Eukleidova algoritmu

Eukleidovým algoritmem počítáme $\gcd(a, b)$, kde $a \geq b > 0$.

- Počet dělení se zbytkem je $O(\mathrm{len}(b))$.
- Hrubý odhad celkového času je $O(\mathrm{len}(b)^2\,\mathrm{len}(a))$.
- Lze dokázat více: Euleidův algoritmus potřebuje čas $O(\mathrm{len}(b)\,\mathrm{len}(a))$.

Rozšířeným Eukleidovým algoritmem spočítáme $\gcd(a, b)$ a dále $s, t \in \mathbb{Z}$ takové, že $sa + tb = \gcd(a, b)$.

- Rozšířený Euleidův algoritmus potřebuje čas $O(\mathrm{len}(b)\,\mathrm{len}(a))$.
- Je-li $\gcd(a, n) = 1$, pak se a^{-1} spočte v čase $O(\mathrm{len}(n)^2)$ rozšířeným Euleidovým algoritmem.

Časová složitost reziduálního počítání

Počítáme s celými čísly a, b, výsledky budou v rozmezí $-M$ až M, respektive 0 až $2M$.

- Zvolíme sadu "malých navzájem nesoudělných čísel", většinou prvočísel $p_1, \ldots, p_k$ tak, aby $n = \prod_{i=1}^{k} p_i > 2M$.
  Všechna prvočísla $p_i < 2^C$, kde C je konstanta.
  Počítat budeme reziduálně pomocí Čínské věty o zbytcích.
- Universální koeficienty q_i, $1 \leq i \leq k$, pro Čínskou větu o zbytcích se spočtou v čase $O(\mathrm{len}(n)^2)$,
  přitom $\mathrm{len}(q_i) \simeq \mathrm{len}(n)$.
  Tyto koeficienty ovšem počítáme pouze jednou!

Časová složitost reziduálního počítání

- Zbytky a_i, b_i pro čísla a, b modulo p_i, $1 \leq i \leq k$, spočteme v čase $O(C\,\mathrm{len}(n)) = O(\mathrm{len}(n))$.
- Aritmetické operace v $\mathbb{Z}_{p_i}$ trvají konstantní čas:
  $a_i \pm b_i$, $a_i \cdot b_i$, resp. a_i^r, a_i^{-1}, je-li $\gcd(a_i, n) = 1, r < p_i$,
  se v každém $\mathbb{Z}_{p_i}$ spočtou v čase nejvýše $O(C^3) = O(1)$.
- Řešení příslušné zbytkové soustavy pro $a \pm b$, $a \cdot b$,
  resp. a^s, a^{-1} v $\mathbb{Z}_n$, je-li $\gcd(a, n) = 1$, je lineární kombinací koeficientů q_i prováděnou v $\mathbb{Z}_n$ a dopočte se v čase $O(kC\,\mathrm{len}(n)) = O(\mathrm{len}(n))$.
- Reziduální počítání (s předpočítanými q_i) funguje v lineárním čase s multiplikativní konstantou kC.

Časová složitost reziduálního počítání

Poznámka

Součin všech prvočísel menších než 2^{16} je přibližně $2^{90\,000}$.
Můžeme reziduálně sčítat a násobit čísla o 45 000 bitů v lineárním čase.
Odhad multiplikativní konstanty: prvočísel do 2^{16} je $k \doteq 5000$, vynásobení zbytků bude v rámci 32−bitového slova v jednotkovém čase. Tedy je to zhruba 9−krát rychlejší než kvadratický čas.

Časová složitost výpočtů modulo n

Literatura

- Velebil: Diskrétní matematika. Kapitola 3.4.
  ftp://math.feld.cvut.cz/pub/velebil/y01dma/dma-notes.pdf
- Shoup: A Computational Introduction to Number Theory and Algebra. Kapitoly 2.4-7, 3.1-4, 4.1-4.
  http://shoup.net/ntb/