Číselné algoritmy

Pollardova $(p-1)$-metoda

15. 3. 2021

$(p-1)$-metoda, úvod

- Představena v článku: J. Pollard: *Theorems for factorization and primality testing*, Proc. Cambridge Phil. Soc. 76(1974), 521 - 528.
- Faktorizační algoritmus pro čísla, které mají dělitele speciálního tvaru
- Princip metody je aplikován rovněž v algoritmu ECM

B-hladká a B-mocná čísla

Definice

Mějme $B, x \in \mathbb{N}$.

a) *Řekneme, že x je B-*hladké *(B-smooth), pokud každý prvočíselný dělitel x je $\leq B$ (tj $\forall p \in \mathbb{P}\ p \mid x \Rightarrow p \leq B$).*

b) *Řekneme, že x je B-*mocné *(B-powersmooth), pokud každá prvočíselná mocnina dělící x je $\leq B$ (tj $\forall p \in \mathbb{P}, \forall e \in \mathbb{N}$ $p^e \mid x \Rightarrow p^e \leq B$).*

Pro $(p-1)$-metodu je důležitý pojem B-mocného čísla.

číslo e_B

Pro $B \geq 2$ je každá mocnina dvojky B-hladká, tedy B-hladkých čísel je nekonečně mnoho. Naproti tomu B-mocných čísel je jen konečně mnoho.

Lemma

Nechť $B \geq 2$. Každé B-mocné číslo x splňuje

$$x \leq e_B := \prod_{p \in \mathbb{P}, p \leq B} p^{\lfloor \log_p(B) \rfloor}$$

Důkaz: Je-li $x = \prod_{i=1}^{k} p_i^{e_i}$ prvočíselná faktorizace x (tedy $p_1, \ldots, p_k \in \mathbb{P}$ po dvou různá prvočísla, $e_1, \ldots, e_k \in \mathbb{N}$), je $p_i^{e_i} \leq B$ pro každé $1 \leq i \leq k$. Tedy $p_i \leq B$ a $e_i \leq \log_{p_i}(B)$, tudíž

$$x \leq e_B$$

Poznámka: Z definice e_B vidíme, že e_B je B-mocné.

Co potřebujeme vědět o e_B

e_B je největší B-mocné číslo při klasickém uspořádání.
Pro $(p-1)$-metodu potřebujeme vědět, že e_B je největší B-mocné číslo též při uspořádání dělitelností.

Lemma

Necht' $B \geq 2$. Každé B-mocné číslo je dělitelem e_B. Tedy e_B je nejmenší společný násobek všech B-mocných čísel.

Důkaz: V předchozím důkazu jsme viděli, že $x \mid e_B$ pro každé B-mocné číslo x.

Idea $(p-1)$-metody

Na vstupu je složené $N \in \mathbb{N}$ (zpravidla předpokládáme, že N není mocnina prvočísla).
Předpokládejme, že existuje $p \in \mathbb{P}$, takové, že $p \mid N$ a $(p-1)$ je B-mocné.
Zvolíme $a \in \mathbb{Z}_N^*$ a spočteme $a^{e_B} \bmod N$.
Pak $\mathrm{NSD}(a, p) = 1$. Podle malé Fermatovy věty je

$$a^{p-1} \equiv 1 \pmod{p}.$$

Protože $p-1$ je B-mocné, je $(p-1) \mid e_B$. Takže pro nějaké $k \in \mathbb{N}$ je $e_B = k(p-1)$.

$$a^{e_B} \bmod p = (a^{p-1})^k \bmod p = 1.$$

Víme tedy, že $p | \mathrm{NSD}(a^{e_B} - 1 \bmod N, N)$. Tento NSD by tedy mohl být vlastní dělitel N.

Nástřel algoritmu Pollard-(p-1)

Vstup: N

Výstup: vlastní dělitel N, nebo neúspěch

K dispozici: B, $p_1, p_2, \ldots, p_k$ posloupnost všech prvočísel $\leq B$

1. Zvol $a \in \{2, \ldots, N-1\}$ náhodně (často též $a = 2$)
2. **if** $\mathrm{NSD}(a, N) > 1$ **then return** $\mathrm{NSD}(a, N)$.
3. Spočteme e_B
4. Spočteme $a := a^{e_B} \bmod N$
5. **if** $1 < \mathrm{NSD}(a-1, N) < N$ **then return** $\mathrm{NSD}(a-1, N)$
   else return Neúspěch

Problémy k vyřešení:

- Neznáme B. Protože neznáme faktorizaci N, musíme B nějak zvolit - například podle výpočetních možností. Případně můžeme zkusit začít s nějakou hodnotou B a pokud nastane neúspěch z důvodu nízkého B, zkusíme štěstí znovu s nějakou vyšší volbou B.
- Přestřelená volba B. Pokud máme $N = pq$, $1 < p < q \in \mathbb{P}$ a $p-1$ i $q-1$ jsou obě B-mocná, vyjde $\mathrm{NSD}(a-1, N) = N$ (tj každá taková volba B povede k neúspěchu). Zkusíme trochu změnit organizaci výpočtu, abychom zvýšili šanci na nalezení vlastního dělitele N.

Algoritmus Pollard-$(p-1)$, první verze

Vstup: $N \in \mathbb{N}$ složené, ne mocnina prvočísla

Výstup: vlastní dělitel N, nebo neúspěch

K dispozici: B, $p_1, p_2, \ldots, p_k$ posloupnost všech prvočísel $\leq B$

1. Zvol $a \in \{2, \ldots, N-1\}$ náhodně (často též $a = 2$)
2. **if** $\mathrm{NSD}(a, N) > 1$ **then return** $\mathrm{NSD}(a, N)$.
3. **for** $i = 1$ to k **do**
   $c_i := \lfloor \log_{p_i}(B) \rfloor$
   $e_i := p_i^{c_i}$
   $a := a^{e_i} \bmod N$ //rychlé mocnění
   if $\mathrm{NSD}(a-1, N) > 1$ **then GOTO** 4.
4. **if** $1 < \mathrm{NSD}(a-1, N) < N$ **then return** $\mathrm{NSD}(a-1, N)$
   else return Neúspěch

Něco ke složitosti

Počet iterací aproximujeme číslem $B/\ln(B)$. Výpočet čísla e_i bychom mohli realizovat takto

$e_i := p_i$
while $(p_i * e_i) \leq B$ **do**
$e_i := e_i * p_i$

Počet iterací while-cyklu $\leq \log(B)$ v každé iteraci násobíme přirozená čísla $\leq B$.
Tento výpočet zvládneme v čase $O(\log^3(B))$.
Umocnění $a := a^{e_i} \mod N$ v čase $O(\log(B)\log^2(N))$.
Výpočet $\mathrm{NSD}(a-1, N)$ v čase $O(\log^2(N))$.
Celkově tak dostaneme odhad složitosti $O(B(\log^2(B) + \log^2(N)))$.
Za celkem rozumného předpokladu $B \leq N$ se dostaneme ke složitosti $O(B\log^2(N))$.

Nejhorší případ

Definice

Prvočíslo p se nazývá bezpečné *(safe), pokud je tvaru $2p' + 1$, kde $p' \in \mathbb{P}$.*

Pokud je $N = pq$ součin dvou různých bezpečných prvočísel, potřebujeme volit $B \geq \min(\frac{p-1}{2}, \frac{q-1}{2})$.

Úspěšnost algoritmu

Pokud je důvodem neúspěchu $\mathrm{NSD}(a-1, N) = 1$ i po projití všech prvočísel, víme, že pro žádné prvočíslo $p \mid N$ není $(p-1)$ B-mocné. Nezbývá než zvětšit B nebo zkusit jinou faktorizační metodu.
Pokud naopak končíme s $\mathrm{NSD}(a-1, N) = N$ je dobrá šance, že N má vhodný prvočíselný faktor. V tomto případě dává smysl algoritmus zkusit pustit znovu s jinou volbou a.
Příklad selhání tohoto typu pro vstup tvaru $N = pq$, $p < q \in \mathbb{P}$ představují prvky $a \in \mathbb{Z}_N^*$, pro které je
$o_{\mathbb{Z}_p^*}(a \bmod p) = o_{\mathbb{Z}_q^*}(a \bmod q)$.

Algoritmus Pollard-$(p-1)$, druhá verze

Vstup: $N \in \mathbb{N}$ složené, ne mocnina prvočísla

Výstup: vlastní dělitel N, nebo neúspěch

K dispozici: B, $p_1, p_2, \ldots, p_k$ posl. všech prvočísel $\leq B$

1. Zvol $a \in \{2, \ldots, N-1\}$ náhodně (často též $a = 2$)
2. **if** $\mathrm{NSD}(a, N) > 1$ **then return** $\mathrm{NSD}(a, N)$.
3. **for** $i = 2$ to k **do**
   $c_i := \lfloor \log_{p_i}(B) \rfloor$
   $e_i := p_i^{c_i}$
   $a := a^{e_i} \bmod N$ //rychlé mocnění
4. **if** $\mathrm{NSD}(a-1, N) > 1$ **then GOTO 6.**
5. **for** $i = 1$ to $\lfloor \log(B) \rfloor$
   $a := a^2 \bmod N$
   if $\mathrm{NSD}(a-1, N) > 1$ **then GOTO 6.**
6. **if** $1 < \mathrm{NSD}(a-1, N) < N$ **then return** $\mathrm{NSD}(a-1, N)$
   else return Neúspěch

Pravděpodobnost úspěchu druhé verze

Tvrzení

Předpokládejme, že $N = pq$, kde $2 < p < q \in \mathbb{P}$, kde $(p-1)$ i $(q-1)$ jsou B-mocná. Pak pravděpodobnost úspěchu druhé verze algoritmu je alespoň $\frac{1}{2}$.

Důkaz: Máme $a \in \mathbb{Z}_N^*$. Označme

$$e'_B = \prod_{p \in \mathbb{P}, 2 < p \leq B} p^{\lfloor \log_p(B) \rfloor}, a' := a^{e'_B} \bmod N\,.$$

Napíšeme $p - 1 = 2^{\ell_p} p'$, $q - 1 = 2^{\ell_q} q'$, p', q' lichá. Máme pak

$$\mathbb{Z}_N^* \simeq ((\mathbb{Z}_{2^{\ell_p}}, +) \times (\mathbb{Z}_{p'}, +)) \times ((\mathbb{Z}_{2^{\ell_q}}, +) \times (\mathbb{Z}_{q'}, +))$$

$$a \mapsto ((\alpha_p, \beta_p), (\alpha_q, \beta_q))$$

$$a' \mapsto ((e'_B \alpha_p, e'_B \beta_p), (e'_B \alpha_q, e'_B \beta_q)) = ((e'_B \alpha_p, 0), (e'_B \alpha_q, 0))$$

K tomu, aby algoritmus skončil úspěchem stačí, aby $o_{\mathbb{Z}_{2^{\ell_p}}}(e'_B \alpha_p) \neq o_{\mathbb{Z}_{2^{\ell_q}}}(e'_B \alpha_q)$. Ekvivalentně $o_{\mathbb{Z}_{2^{\ell_p}}}(\alpha_p) \neq o_{\mathbb{Z}_{2^{\ell_q}}}(\alpha_q)$. Stejným argumentem jako na první přednášce ukážeme, že náhodně volený prvek $\mathbb{Z}_N^*$ má tuto vlastnost s pravděpodobností $\geq \frac{1}{2}$.

Dvoufázová $(p-1)$-metoda

Důležité vylepšení algoritmu představuje přidání druhé fáze. Účinnost algoritmu se rozšíří na vstupy dělitelné prvočíslem p, pro které je $(p-1)$ součin malých prvočísel a jednoho většího. Rozšíření využívá, že rozdíly mezi dvěma po sobě jdoucími prvočísly jsou relativně malé.

Popis dvoufázové metody

Vstup: $N \in \mathbb{N}$ složené, ne mocnina prvočísla

Výstup: vlastní dělitel N, nebo neúspěch

K dispozici: $B < B_1$, $2 = p_1, p_2, \ldots, p_\ell$ posl. všech prvočísel $\leq B_1$, $p_1, \ldots, p_k$ je posl. všech prvočísel $\leq B$

1. Zvol $a \in \{2, \ldots, N-1\}$ náhodně (často též $a = 2$)

2. **if** $\mathrm{NSD}(a, N) > 1$ **then return** $\mathrm{NSD}(a, N)$.

3. Spočti $b := a^{e_B} \bmod N$, druhá fáze pokračuje, pouze pokud $\mathrm{NSD}(b-1, N) = 1$.

4. Z posloupnosti $p_{k+1}, p_{k+2}, \ldots, p_\ell$ spočti $c_i := p_{k+i+1} - p_{k+i}$, ulož hodnoty $b_i := b^{c_i} \bmod N$

5. $b := b^{p_{k+1}} \bmod N$

6. **for** $i = 1$ to $\ell - k - 1$

  if $\mathrm{NSD}(b-1, N) > 1$ **then GOTO 7.**

  $b := b * c_i \bmod N$

7. Vyhodnocení $\mathrm{NSD}(b-1, N)$

Konec 3. přednášky

Děkuji za pozornost.