Číselné algoritmy

Dixonova faktorizace

26. 4. 2021

Dixonova faktorizace úvod

- John D. Dixon: *Asymptotically fast factorization of integers*, Mathematics of Computation, 1981
- Analýza algoritmu vede (při určité volbě parametrů) k subexponenciálnímu faktorizačnímu algoritmu
- V naší přednášce Dixonův algoritmus chápeme jako schéma z něhož vycházejí algoritmy CFRAC a kvadratické síto

Fermatova faktorizace

Připomeňme Fermatovu metodu, která hledala pro $N \in \mathbb{N}$ čísla x, y tak, aby $N = x^2 - y^2$. Z tohoto vyjádření pak máme faktorizaci N:

$$N = (x - y)(x + y)\,.$$

Zkusme tuto podmínku zeslabit na $x^2 \equiv y^2 \pmod{N}$. Pak dostaneme

$$(x - y)(x + y) \equiv 0 \pmod{N}$$

Musí proto platit $\mathrm{NSD}(x - y, N) > 1$ nebo $\mathrm{NSD}(x + y, N) > 1$.
Kongruence $x^2 \equiv y^2 \pmod{N}$ může platit z triviálních důvodů, tj pokud platí $x \equiv \pm y \pmod{N}$. V takovém případě bude vždy $\mathrm{NSD}(x - y, N) = N$ nebo $\mathrm{NSD}(x + y, N) = N$.
Naopak, bude-li $x \not\equiv \pm y \pmod{N}$, bude jistě $\mathrm{NSD}(x - y, N) < N$ a $\mathrm{NSD}(x + y, N) < N$, a tedy $\mathrm{NSD}(x - y, N)$ nebo $\mathrm{NSD}(x + y, N)$ bude vlastní dělitel N.

Relace, hlavní princip Dixonovy metody

Necht' $N \in \mathbb{N}$. Uspořádaná dvojice $(x, y) \in \mathbb{Z} \times \mathbb{Z}$ je *relace*, pokud platí

$$x^2 \equiv y \pmod{N}.$$

Dixonova metoda se hledá relace $(x_1, y_2), (x_2, y_2), \ldots, (x_k, y_k)$ tak, aby

$$\prod_{i=1}^{k} y_i = y^2$$

pro nějaké $y \in \mathbb{Z}$. Pro $x := \prod_{i=1}^{k} x_i$ pak bude platit

$$x^2 = \prod_{i=1} x_i^2 \equiv \prod_{i=1}^{k} y_i = y^2 \pmod{N}$$

Máme tedy čtverce kongruentní modulo N, které se pokusíme využít k faktorizaci N.

Generátor relací

Každý algoritmus založený na Dixonově metodě potřebuje generátor relací.

- Jednoduchým příkladem může být
  1. Zvol náhodně $x \in \mathbb{Z}_N$
  2. return $(x, x^2 \bmod N)$
- Generátor pro kvadratické síto
  1. Zvol $x \in \mathbb{Z}$ poblíž $\sqrt{N}$
  2. return $(x, x^2 - N)$
- Trochu komplikovanější generátor relací má algoritmus CFRAC, relace vznikají při výpočtu řetězového rozvoje $\sqrt{N}$

Faktorizační báze a hladké relace

Algoritmus volí *faktorizační bázi*. Typicky jde o množinu

$$B = \{-1, 2, 3, 5, 7, \ldots, p_{max}\}$$

Relace (x, y) se nazývá *hladká* pokud je y součin prvků faktorizační báze, tedy

$$y = \prod_{b \in B} b^{e_b}$$

Pokud víme, že generátor relací vrací relace tvaru (x, y), kde $y \geq 0$, nemá smysl dávat do faktorizační báze prvek -1.

Poznámka k volbě faktorizační báze

Podobně pokud víme, že prvočíslo nebude dělit pravou stranu žádné relace, nemá smysl dávat ho do faktorizační báze.
Příklad: Uvažme následující generátor relací: Vyber $x \in \mathbb{Z}$, vrať $(x, x^2 - N)$. Pokud N není čtverec celého čísla, bude $x^2 - N \neq 0$ pro každé $x \in \mathbb{Z}$. Pokud $p \in \mathbb{P}$, existuje $x \in \mathbb{Z}$ takové, že $p \mid x^2 - N$ pouze pokud N je kvadratické reziduum modulo p. Pro takový generátor relací má bychom do faktorizační báze dali pouze prvočísla splňující $\left(\frac{N}{p}\right) = 1$.

Další poznámky k volbě faktorizační báze

Z nagenerované množiny relací vybereme hladké relace. Hladkost relace můžeme testovat zkusmým dělením (případně sofistikovanější metodou, například u kvadratického síta dochází k zásadnímu zrychlení algoritmu přidáním prosívací fáze). Volba faktorizační báze by měla zohledňovat tyto aspekty:

- četnost výskytu hladkých relací
- výpočetní náročnost testování hladkosti relace

Faktorizační báze z Dixonova článku obsahovala všechna prvočísla nepřesahující $e^{\sqrt{2\ln(N)\ln(\ln(N))}}$.

Lineární fáze

Předpokládejme, že jsme nalezli hladké relace $(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)$. Úkolem lineární fáze je vybrat z těchto relací podmnožinu takovou, že součin pravých stran vybraných relací bude čtverec.
Jinými slovy, hledáme $I \subseteq \{1, 2, \ldots, k\}$ (neprázdnou), pro kterou bude existovat $y \in \mathbb{Z}$ tak, že

$$\prod_{i \in I} y_i = y^2$$

Sestavme matici A, která bude mít k řádků a $|B|$ sloupců. Sloupce této matice budeme indexovat prvky B.
Protože relace jsou hladké, máme

$$y_i = \prod_{b \in B} b^{e_{i,b}}$$

Do matice A na pozici (i, b) napíšeme prvek $e_{i,b} \bmod 2$.

Lineární fáze 2

Budeme hledat řádkové vektory $u = (u_1, u_2, \ldots, u_k) \in \mathbb{Z}_2^k$, pro které je

$$uA \equiv 0 \pmod 2$$

Tedy jde o řešení soustavy homogenních lineárních rovnic nad dvouprvkovým tělesem.
Je-li $u = (u_1, u_2, \ldots, u_k)$ řešení takovéto soustavy, platí pro každé $b \in B$

$$\sum_{i=1}^{k} u_i e_{i,b} \equiv 0 \pmod 2 .$$

Položme $I_u := \{i \in \{1, \ldots, k\} \mid u_i = 1\}$.
Pak $\prod_{i \in I_u} y_i = \prod_{b \in B} b^{\sum_{i \in I_u} e_{i,b}} = \prod_{b \in B} b^{\sum_{i=1}^{k} u_i e_{i,b}}$. Položme $e_b = \frac{1}{2} \sum_{i=1}^{k} u_i e_{i,b}$ a $y_u = \prod_{b \in B} b^{e_b}$. Pak je $\prod_{i \in I_u} y_i = y_u^2$, a tedy

$$(\prod_{i \in I_u} x_i)^2 \equiv y_u^2 \pmod N$$

Dobrá a špatná řešení

Ne každé řešení poskytnuté lineární fází lze použít pro faktorizaci vstupu.
Řekneme, že řešení $u \in \mathbb{Z}_2^k$ je *špatné*, pokud je

$$(\prod_{i \in I_u} x_i) \equiv \pm y_u \pmod{N}$$

Za určitých předpokladů (například pokud N je součin dvou různých lichých prvočísel) lze dokázat, že špatná řešení tvoří podprostor $\mathbb{Z}_2^k$.
Obecně nemáme mechanismus, který by zaručil, že existuje alespoň jedno řešení, které není špatné. Nicméně, pokud takové řešení existuje, lze alespoň jednu polovinu všech řešení použít k úspěšné faktorizaci N.
V praxi se proto lineární fáze snaží najít více lineárně nezávislých řešení.

Lineární fáze - závěrečné poznámky

Pro větší vstupy je třeba brát velké faktorizační báze. Provedení lineární fáze Gaussovou eliminační metodou by pak mohlo narazit na paměťovou i časovou náročnost algoritmu.
Ve sloupci matice indexovaném prvočíslem b očekáváme nenulové pouze každé b-té pole. Většina polí matice je proto nulová. Faktorizační algoritmy proto většinou využívají specializované algoritmy pro řešení soustav rovnic s řídkou maticí (Wiedemannova bloková metoda, Lanczosova bloková metoda).

Parciální relace

Hladké relace jsou poměrně vzácné. Většina algoritmů proto pracuje i s parciálními relacemi. To jsou relace (x, y), kde pravá strana je součin prvků z faktorizační báze a několika málo (typicky jednoho nebou dvou) prvočísel mimo faktorizační bázi.
Příklad: Máme faktorizační bázi $B = \{-1, 2, 3, \ldots, p_{\max}\}$ a zvolíme $p_{\max} < q < p_{\max}^2$. Dostaneme relaci (x, y). Pravou stranu relace podělíme prvočísly z faktorizační báze v nejvyšších možných mocninách. Dostaneme tak

$$y = \pm \prod_{p \in B \cap \mathbb{P}} p^{v_p(y)} . y' \,,$$

kde $y' \in \mathbb{N}$ není dělitelné prvočíslem z faktorizační báze. Pokud je $y' = 1$, máme hladkou relaci. Pokud je $1 < y' < q$, je nutně $y' \in \mathbb{P} \setminus B$. Dostaneme pak parciální relaci s jedním velkým prvočíslem y'.

Malý koeficient(*)

Malý koeficient je metoda, která se snaží zvýšit výtěžnost malých prvočísel ve faktorizační bázi.
Základní idea: Místo N faktorizujeme kN, kde $k \in \mathbb{N}$ a doufáme, že výstup algoritmu poskytne také netriviální faktor N. Volbu k se snažíme nastavit tak, aby testování hladkosti relací bylo v průměru co nejrychlejší.
Pro generátor relací tvaru $(x, x^2 - N)$ bychom mohli postupovat takto: Vybereme množinu prvočísel S (malá lichá prvočísla, u kterých chceme, aby často dělila pravé strany relací). Pro $k \in \mathbb{N}$ (typicky malé bezčtvercové) určíme přínos k dle vzorce

$$-\log(k) + \sum_{p \in S, p|k} \frac{\log(p)}{p-1} + \sum_{p \in S, \left(\frac{kN}{p}\right)=1} \frac{2\log(p)}{p-1}$$

Jako malý koeficient pro algoritmus pak použijeme to k, u kterého vyšel největší přínos.

Konec osmé přednášky

Děkuji za pozornost.