1 Úvod

Práca sa zaoberá AKS-algoritmom. Tento algoritmus je prvý známy polynomiálny deterministický algoritmus na testovanie prvočíselnosti.

2 Testovanie prvočíselnosti

2.1 Známe algoritmy

Hľadanie prvočísel a zisťovanie, či dané číslo je prvočíslo, sa vyskytovalo medzi teoreticko-číselnými algoritmami už dávno a má aj veľké praktické využitie - napríklad v kryptológii. Je snaha nájsť čo najefektívnejší algoritmus na testovanie prvočíselnosti.

Spomenul by som ešte, že pri algoritmoch operujúcich s číslami sa časová zložitosť vyjadruje v závislosti od dĺžky zápisu vstupu - čiže od počtu cifier. O polynomiálnom algoritme teda môžeme hovoriť vtedy, ak sa jeho časová zložitosť dá vyjadriť ako polynóm od $\lg n$.

Na úvod by som spomenul niektoré známe algoritmy, ktoré boli používané na zisťovanie prvočíselnosti. Najjednoduchším algoritmom, ktorý si môžeme predstaviť, je Eratostenovo sito známe už zo základnej školy. Ide jednoducho o test, či je číslo deliteľné nejakým menším číslom. Z novších a efektívnejších testov môžeme spomenúť napríklad Lucas-Lehmerov test, ktorý sa dá použiť na hľadanie Mersennových prvočísel - prvočísel tvaru $2^p - 1$. Rabin-Millerov test je náhodnostný test - v prípade, že vráti odpoveď TRUE, dané číslo je s vysokou pravdepodobnosťou prvočíslo, pričom túto pravdepodobnosť môžeme zvoliť ľubovoľne malú. Za predpokladu rozšírenej Riemannovej hypotézy je to deterministický polynomiálny algoritmus. Medzi moderné algoritmy patrí ECPP - elliptic curves primality proving; testovanie prvočíselnosti pomocou eliptických kriviek a APR algoritmus, resp. jeho modifikácia APRCL. Posledný algoritmus je „skoro polynomiálny“, keďže jeho časová zložitosť sa dá vyjadriť ako $\lg n$ umocnený na exponent, ktorým je veľmi pomaly rastúca funkcia.

O žiadnom z doteraz spomínaných algoritmov sa nepodarilo ukázať, že je deterministický a polynomiálny - prvý takýto algoritmus našli v roku 2002 indickí matematici Agrawal, Kayal a Saxena. Ide práve o AKS-algoritmus, ktorým v tejto práci budeme zaoberať.

2.2 AKS-algoritmus

Aspoň v stručnosti porovnajme prednosti aj nevýhody AKS-algoritmu v porovnaní s algoritmami, ktoré sme spomenuli doteraz.

Už sme spomenuli, že tento algoritmus je prvým známym bezpodmienečne polynomiálnym a deterministickým algoritmom na testovanie prvočíselnosti. (Pod výrazom „bezpodmienečne“ rozumieme to, že na dôkaz jeho polynomiálnej časovej zložitosti nepotrebujeme prijať žiadnu dodatočnú teoreticko-číselnú podmienku.) Tento poznatok bol prekvapivý pre mnohých odborníkov zaoberajúcich sa touto oblasťou. Ďalšou výhodou je podstatne jednoduchší matematický aparát potrebný na dôkaz jeho správnosti a časovej zložitosti. Aj bez toho, že by sme museli podrobne študovať, aké výsledky sú použité v jednotlivých algoritmoch, ľahko sa o tom môžeme presvedčiť aj jednoducho porovnaním dĺžky článkov, v ktorých sú spomínané algoritmy zavedené. Zatiaľčo článok popisujúci AKS-algoritmus má 9 strán, články týkajúce sa ostatných algoritmov majú okolo 30 strán.

Bohužiaľ, hoci ide o polynomiálny algoritmus, a teda asymptoticky rýchlejší, v praxi výrazne zaostáva na za používanými algoritmami. Takisto algoritmy, ktoré sú pravdepodobnostné, sa dajú v praxi použiť. Buď je pravdepodobnosť chyby natoľko malá, že ju môžeme zanedbať, alebo môžeme použiť algoritmy, ktoré generujú „svedka" - číslo, pomocou ktorého možno overiť prvočíselnosť podstatne rýchlejšie než pomocou samotného algoritmu. Napriek tomu je snaha vylepšovať AKS-algoritmus alebo hľadať efektívne algoritmy založené na podobných úvahách ako tento algoritmus. Zdá sa, že je tu ešte dostatočne veľký priestor na vylepšovanie - neskôr keď sa dostaneme k popisu algoritmu povieme si, že časovo najnáročnejším krokom je overovanie polynomiálnych kongruencií, čiže časová zložitosť závisí hlavne od počtu kongruencií, ktoré treba overiť. Na základe niektorých heuristických úvah ako aj výpočtových pokusov sa zdá, že by stačilo overovať aj podstatne menej kongruencií.

Naše porovnanie môžeme stručne zhrnúť tak, že AKS-algoritmus má (v súčasnej podobe) oveľa väčší teoretický než praktický význam.

3 AKS-algoritmus

3.1 Správnosť algoritmu

Základom viacerých algoritmov na testovanie prvočíselnosti je známa Malá Fermatova veta.

Veta 3.1 (Malá Fermatova veta). *Nech p je prvočíslo. Potom*

$$a^p \equiv a \pmod{p} \tag{1}$$

pre každé celé číslo a.

AKS-algoritmus používa isté zovšeobecnenia tejto vety. Nevýhodou Malej Fermatovej je to, že aj mnoho zložených čísel spĺňa vzťah (1). Tento nedostatok už nemá nasledujúca veta.

Veta 3.2. *Nech $n \geq 2$ a $a \geq 0$ sú celé čísla.*

(i) Ak n je prvočíslo, tak v okruhu $\mathbb{Z}_n[x]$ platí rovnosť

$$(x+a)^n = x^n + a.$$

(ii) Ak $(a, n) = 1$ a n nie je prvočíslo tak

$$(x + a)^n \neq x^n + a.$$

v okruhu polynómov $\mathbb{Z}_n[x]$.

Použitie tejto vety by však viedlo k exponenciálnemu algoritmu, pretože by sme museli vyrátať všetkých n koeficientov polynómu $(x + a)^n$. Nasledujúce vylepšenie tejto vety nám umožní rátať len polynomiálne veľa koeficientov - a (zjednodušene povedané) práve vďaka tomu je AKS-algoritmus polynomiálny.

Veta 3.3. *Nech $n \geq 2$ je prirodzené číslo, p a r sú prvočísla, p delí n a $\ell = \lfloor 2\sqrt{r} \lg n \rfloor + 1$. Multiplikatívny rád čísla n modulo r označme d. Nech platia nasledujúce predpoklady:*

(i) $(r, n) = 1$,

(ii) $p > r$,

(iii) $d > 4 \lg^2 n$,

(iv) pre každé a také, že $1 \leq a \leq \ell$ platí v $\mathbb{Z}_n[x]$

$$(x - a)^n = (x^n - a) \mod (x^r - 1).$$

Potom n je mocnina prvočísla p.

3.2 Popis algoritmu

Teraz v stručnosti popíšeme AKS-algoritmus a povieme si niečo o tom, ako jeho správnosť vyplýva zo spomínanej vety.

1. Ak $n = a^b$ pre nejaké prirodzené čísla $a, b > 1$, tak vrátime `FALSE`.

2. Hľadáme najmenšie prvočíslo r také, že platí buď $(n, r) > 1$ alebo $(n, r) = 1$ a multiplikatívny rád d čísla n modulo r je viac než $4 \lg^2 n$.

3. Ak $r = n$, tak vrátime `TRUE`.

4. Ak $(n, r) > 1$ tak vrátime `FALSE`.

5. Postupne pre a od 1 po $\ell = \lfloor 2\sqrt{r} \lg n \rfloor + 1$ testujeme rovnosť

   $$(x - a)^n = x^n - a \mod (x^r - 1)$$

   v $\mathbb{Z}_n[x]$. Akonáhle nájdeme najmenšie a, pre ktoré táto rovnosť neplatí, vrátime `FALSE`.

6. Ak rovnosť platí pre všetky $a = 1, \ldots, \ell$, tak vrátime `TRUE`.