SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1590

Problem PRIMES

Hrvoje Bandov

Zagreb, lipanj 2010.

Sadržaj

1. Uvod

Teorija složenosti uz izračunljivost i automate čini jedno od tri usko povezana područja teorijskog računarstva. Teorija brojeva je veliko područje matematike koje se bavi proučavanjem cijelih brojeva. Dugo vremena je smatrano kao "čisti" dio matematike bez praktičnih primjena, no danas saznanja iz teorije brojeva čine temelj moderne kriptografije o čijoj praktičnoj važnosti ne treba mnogo govoriti.

Presjek teorijskog računarstva i teorije brojeva bi mogli staviti pod jedinstveni naziv: računalna teorija brojeva (engl. *computational number theory*). Posebno nam je zanimljivo proučavanje složenosti nekih problema iz teorije brojeva, npr. ispitivanje primarnosti prirodnog broja, rastavljanje broja na proste faktore, generiranje prim brojeva itd.

U ovom radu bavit ćemo se problemom ispitivanja prim (prostih) brojeva (određivanja je li zadani prirodni broj primaran ili složen). Naivni algoritam bi mogao ispitivati djeljivost sa svim brojevima manjim od $\sqrt{n}$, no takav algoritam nije efikasan jer eksponencijalno ovisi o broju znamenki (ili bitova) kojima je n zapisan.

Logično je da se mnogi rezultati o složenosti problema teorije brojeva temelje na spoznajama i znanju same teorije brojeva koje čine bazu mnogih algoritama.

Započet ćemo sa osnovnim definicijama i teoremima iz teorije složenosti odnosno teorije brojeva, zatim ćemo razmotriti povijesne rezultate o složenosti našeg problema i na kraju izložiti dokaz Agrawala, Kayala i Saxena iz 2004. koji je konačno svrstao problem u klasu P tj. pokazali su da je moguće ispitati primarnost prirodnog broja u polinomnom vremenu.

2. Osnovne definicije i teoremi

Definirat ćemo klase složenosti P i NP. Koristit ćemo formalno Turingov stroj (skraćeno TS) kao model računala ali ga nećemo nikad "programirati". Koristimo se asimptotskom analizom i notacijom "veliko O". Uvodne definicije koje nećemo iznositi mogu se naći u svakom uvodniku u teorijsko računarstvo poput (Sipser, 2005).

Iz teorije brojeva pretpostavljamo bliskost s nekim osnovnim pojmovima i činjenicama kao što su djeljivost, primarni brojevi, jedinstveni rastav broja na proste faktore itd.

2.1. Složenost

Definicija 2.1. Neka je $f : \mathbb{N} \to \mathbb{N}$ funkcija. $\mathsf{DTIME}(f)$ je skup svih jezika koji su odlučivi na *deterministič kom* TS vremenske složenosti $\mathcal{O}(f)$. $\mathsf{NTIME}(f)$ je skup svih jezika koji su odlučivi na *nedeterminističkom* TS vremenske složenosti $\mathcal{O}(f)$.

Definicija 2.2. Klasa P je skup svih jezika odlučivih na vremenski polinomnom determinističkom TS. Odnosno,

$$\mathsf{P} = \bigcup_k \mathsf{DTIME}\left(n^k\right).$$

Probleme koji se nalaze unutar P obično smatramo za one koje možemo efikasno izračunavati. Primjerice, jezik RELPRIMES $= \{(a, b) \mid a \text{ i } b \text{ su relativno prosti}\}$ se nalazi u P što je posljedica poznatog Euklidovog algoritma.

Naziv klase NP dolazi od *nondeterministic polynomial time* i možemo ga definirati na sličan način kao i P:

$$\mathsf{NP} = \bigcup_k \mathsf{NTIME}\left(n^k\right).$$

Takva definicija nam ne daje slikoviti opis te važne klase. Postoji drugi način na koji opisujemo klasu NP. Neformalno, za svaki element jezika iz NP postoji

kratki dokaz (certifikat) da je u tom jeziku koji možemo *brzo* provjeriti. Pri tom "brzo" znači u polinomnom vremenu, a "kratko" znači duljine polinomno ovisne o duljini elementa jezika. Formalno to zapisujemo na sljedeći način.

Teorem 2.1. *Neka je* L *proizvoljan jezik. Tada je* $\mathsf{L} \in \mathsf{NP}$ *ako i samo ako postoji relacija* $R(w, c) \in \mathsf{P}$ *takva da*

$$\mathsf{L} = \left\{ w \mid (w, c) \in R \text{ za neki } c \text{ i vrijedi } |c| = \mathcal{O}\left(|w|^k\right) \right\}.$$

Dokaz se može pronaći u (Papadimitriou, 1995) i (Sipser, 2005).

Definicija 2.3. Za klasu složenosti C definiramo $\mathsf{co}\mathrm{C} = \{\overline{\mathsf{L}} \mid \mathsf{L} \in \mathrm{C}\}$.

Primijetimo da za svaku deterističku klasu C vrijedi $\mathrm{C} = \mathsf{co}\mathrm{C}$ (npr. $\mathsf{P} = \mathsf{coP}$) jer je dovoljno obrnuti prihvaćanje TS-a (kad jedan TS prihvaća niz drugi ga jednostavno odbije i obrnuto).

Nedeterministički TS prihvaća niz ako *bar jedna* od grana u stablu računanja prihvaća niz, a odbija ako *sve* odbijaju. Zbog te asimetrije ne možemo jednostavno obrnuti odluku nedeterminističkog TS-a i time je pitanje je li $\mathsf{NP} = \mathsf{coNP}$ puno teže i odgovor nam još nije poznat iako smo skloni vjerovati da je $\mathsf{NP} \neq \mathsf{coNP}$. $\mathsf{P} = \mathsf{NP}$ bi očito impliciralo $\mathsf{NP} = \mathsf{coNP}$.

2.2. Teorija brojeva

Najveći zajednički djeljitelj brojeva a i b označavamo s $\mathrm{nzd}\,(a, b)$. Eulerovu funkciju koja nekom broju n pridružuje broj svih brojeva manjih od n koji su relativno prosti s n označavamo s $\phi(n)$. Krenut ćemo od definicije kongruencije, jednog od osnovnih pojmova u teoriji brojeva.

Definicija 2.4. Ako cijeli broj $m \neq 0$ dijeli razliku $a - b$ (pišemo $m|(a-b)$), onda kažemo da je a kongurentan b modulo m i pišemo $a \equiv b \pmod{m}$.

Relacija "biti kongurentan modulo m" je relacija ekvivalencije. Iznjet ćemo još neka jednostavna svojstva bez dokazivanja.

Propozicija 2.1. Neka su a, b, c i d cijeli brojevi.

1. Ako je $a \equiv b \pmod{m}$ i $c \equiv d \pmod{m}$, onda je $a + c \equiv b + d \pmod{m}$, $ac \equiv bd \pmod{m}$.
2. Ako je $a \equiv b \pmod{m}$ i $d|m$, onda je $a \equiv b \pmod{d}$.

3. Ako je $a \equiv b \pmod{m}$, onda je $ac \equiv bc \pmod{mc}$ za svaki $c \neq 0$.

Kako je riječ o relaciji ekvivalencije onda možemo promatrati i klase ekvivalencije modulo m. Označavamo ih s $[a]_m$ (ili skraćeno $[a]$ ako je m jasan iz konteksta), a po definiciji vrijedi $b \in [a]$ ako i samo ako je $a \equiv b \pmod{m}$. Svaki element iz $[a]$ nazivamo reprezentantom tog razreda.

Možemo definirati operacije zbrajanja i množenja nad klasama ekvivalencije:

$$[a] + [b] = [c] \iff a + b \equiv c \pmod{m},$$

$$[a] \cdot [b] = [c] \iff ab \equiv c \pmod{m}.$$

Skup $\mathbb{Z}_n$ definiramo kao skup svih klasa ekvivalencije modulo n i nad njim su definirane operacije množenja i zbrajanja.

Dokažimo jednu jednostavnu lemu koju ćemo koristiti u dva navrata.

Lema 2.1. *Ako je p prim broj onda p dijeli $\binom{p}{k}$ za sve $0 < k < p$ tj. vrijedi*

$$\binom{p}{k} \equiv 0 \pmod{p}, \quad 0 < k < p.$$

Dokaz. Po definiciji binomnog koeficijenta imamo

$$\binom{p}{k} = \frac{p(p-1)\dots(p-k+1)}{k(k-1)\dots 2\cdot 1}.$$

Svi članovi u nazivniku su manji od p u brojniku i ne dijele p jer je p prim broj. Stoga u $\binom{p}{k}$ postoji faktor p. □

Sad dolazimo do jednog važnog teorema poznatog pod nazivom Fermatov *mali* teorem. Dokazat ćemo ga indukcijom jer na taj način izbjegavamo mnoge prepreke koje inače vode do dokaza.

Teorem 2.2 (Fermatov mali teorem)**.** *Ako je p prim broj onda p dijeli $a^p - a$ za sve $a \in \mathbb{N} \cup \{0\}$ tj. vrijedi:*

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

Dokaz. Za $a = 0$ tvrdnja vrijedi i to je baza indukcije. Pretpostavimo da tvrdnja vrijedi za a i dokažimo da onda vrijedi i za $(a+1)$ odnosno $(a+1)^p \equiv a + 1 \pmod{p}$.

$$\begin{aligned}(a+1)^p &\equiv a^p + \binom{p}{p-1}a^{p-1} + \ldots + \binom{p}{1}a + 1 \pmod{p} \\ &\overset{(1)}{\equiv} a^p + 1 \pmod{p} \\ &\overset{(2)}{\equiv} a + 1 \pmod{p}\end{aligned}$$

Jednakost (1) slijedi iz Leme 2.1, a (2) iz pretpostavke indukcije. □

2.3. Algebra

Grupe nisu u središtu rada pa se nećemo zamarati nekim općenitim svojstvima grupa već ćemo se usredotočiti na ona svojstva koja se tiču jedne specifične grupe.

Definicija 2.5. Abelovom grupom nazivamo skup G zajedno s binarnom operacijom $+$ nad G takvom da vrijedi:

1. za sve $a, b, c \in G$, $a + (b + c) = (a + b) + c$ (tj. $+$ je *asocijativna*)
2. za sve $a, b \in G$, $a + b = b + a$ (tj. $+$ je *komutativna*)
3. postoji $e \in G$ (kojeg zovemo *neutralni element*) takav da za sve $a \in G$, $a + e = a = e + a$
4. za svaki $a \in G$ postoji $a' \in G$ (kojeg zovemo *inverz*) takav da je $a + a' = e = a' + a$

Red grupe je broj elemenata u G i označavamo ga s $|G|$, grupa je beskonačnog reda ako je G beskonačan skup.

Definicija 2.6. Podgrupa grupe $(G, +)$ je grupa $(H, +)$ gdje je H podskup od G.

Primjer 2.1. *Skup $\mathbb{Z}_n$ s operacijom zbrajanja čini Abelovu grupu. Za zbrajanje nad $\mathbb{Z}_n$ vrijede asocijativnost i komutativnost, neutralni element je $[0]_n$, a inverz od $[a]_n$ je $[-a]_n$. Red grupe je n.*

Općenito skup $\mathbb{Z}_n$ i operacija množenja ne čine Abelovu grupu jer nemaju svi elementi inverz!

Primjer 2.2. *Ako je p prim broj onda $\mathbb{Z}_p \backslash \{[0]_p\}$ i operacija množenja čine Abelovu grupu! Za množenje nad $\mathbb{Z}_p$ vrijede asocijativnost i komutativnost, neutralni element je $[1]_p$, a inverz od $[a]_p$ je $[a^{p-2}]_p$ što je posljeica Fermatovog malog teorema:*

$$[a]_p \cdot [a^{p-2}]_p = [aa^{p-2}]_p = [a^{p-1}]_p = [1]_p.$$

Podskup od $\mathbb{Z}_n$ elemenata koji imaju multiplikativni inverz označit ćemo s $\mathbb{Z}_n^*$. Skup $\mathbb{Z}_n^*$ čine svi elementi čiji su reprezentanti relativno prosti s n, no to nećemo dokazivati.

Obično elemente iz $\mathbb{Z}_n$ jednostavno označavamo s nekim reprezentantom a iz $[a]_n$.

Uvest ćemo još neke konvencije. Ako koristimo + kao simbol za operaciju u grupi onda skraćeno zapisujemo:

$$\overbrace{a+a+a+\ldots+a}^{k \text{ puta}} = ka, \quad k \in \mathbb{N}$$

a ako koristimo $\cdot$ kao simbol onda skraćeno zapisujemo:

$$\overbrace{a \cdot a \cdot a \ldots a}^{k \text{ puta}} = a^k, \quad k \in \mathbb{N}.$$

Uobičajeno je da kad koristimo simbol zbrajanja (+) grupu nazivamo *aditivnom*, a kad koristimo simbol množenja ($\cdot$) onda ju nazivamo *multiplikativnom*. U multiplikativnoj grupi inverz od a označavamo s a^{-1}, a neutralni element s 1. U aditivnoj grupi neutralni element obično označavamo s 0. Također definiramo:

$$a^{-k} := (a^{-1})^k \quad \text{odnosno} \quad -ka := k(-a)$$

Definicija 2.7. Neka je X podskup od G grupe $(G, \cdot)$. Onda s $\langle X \rangle$ označavamo najmanju podgrupu od $(G, \cdot)$ koja sadrži X. Nazivamo je podgrupom generiranom s X i vrijedi:

$$\langle X \rangle = \{g_1^{a_1} \ldots g_l^{a_l} \mid g_i \in X, a_i \in \mathbb{Z}, l \geq 1\}$$

Definicija 2.8. Za grupu $(G, \cdot)$ kažemo da je ciklička grupa ako postoji $a \in G$ takav da vrijedi

$$G = \{a^k \mid k \in \mathbb{Z}\} \quad \text{tj.} \quad G = \langle a \rangle.$$

Element a nazivamo generatorom grupe G.

Lako je vidjeti da za svaki element a cikličke grupe postoji $k \in \mathbb{Z}$ takav da je $a^k = 1$. Najmanji takav k nazivamo red elementa a.

Primjer 2.3. *$\mathbb{Z}_5^*$ je ciklička grupa reda 4.*

a	a^2	a^3	a^4
1	1	1	1
2	4	3	1
3	4	2	1
4	1	4	1

Vidimo da su elementi 2 i 3 reda 4 i oba generiraju grupu, a element 4 je reda 2.

Definicija 2.9. Red elementa a u $\mathbb{Z}_n^*$ označavamo s $o_n(a)$.

Definicija 2.10. Komutativni prsten s jedinicom je skup R zajedno s dvije binarne operacije, zbrajanjem $(+)$ i množenjem $(\cdot)$, takve da vrijedi:

1. skup R zajedno s operacijom zbrajanja čini Abelovu grupu (neutralni element označavamo s 0_R)
2. množenje je asocijativno tj. za sve $a, b, c \in R$ vrijedi $a(bc) = (ab)c$
3. za množenje vrijedi distributivnost nad zbrajanjem tj. za sve $a, b, c \in R$ vrijedi $a(b + c) = ab + ac$
4. postoji jedinični element 1_R s obzirom na množenje tj. za sve $a \in R$ vrijedi $1_R \cdot a = a = a \cdot 1_R$
5. množenje je komutativno tj. za sve $a, b \in R$ vrijedi $ab = ba$

U nastavku ćemo uvijek pod *prsten* misliti na *komutativni prsten s jedinicom.*

Definicija 2.11. Prsten F u kojem za svaki element $a \neq 0_F$ postoji multiplikativni inverz nazivamo polje.

Jedan od prstena koji ćemo promatrati je prsten polinoma. Elementi u tom prstenu su polinomi oblika:

$$a_n X^n + a_{n-1} X^{n-1} + \ldots + a_1 X + a_0$$

gdje su $a_n, \ldots, a_0$ koeficijenti iz nekog drugog prstena (npr. $\mathbb{Z}$ ili $\mathbb{Z}_n$). Koeficijent $a_n \neq 0$ nazivamo vodećim koeficijentom i kažemo da je polinom n-tog stupnja. Prsten polinoma kojem su koeficijenti iz prstena R označavamo s $R[X]$.

Operacije množenja i zbrajanja su standardno zadane kao što smo navikli za polinome. Također, kao i u $\mathbb{Z}$ tako i ovdje možemo definirati dijeljenje, ostatak i kongruencije nad polinomima i mnogo toga vrijedi jednako kao u $\mathbb{Z}$ iako to nećemo dokazivati.

Kažemo da je polinom ireducibilan ako ga nije moguće rastaviti na umnožak dvaju ili više polinoma stupnja većeg od jedan.

3. Ispitivanje primarnosti prirodnih brojeva

Definicija 3.1. Jezik PRIMES definiramo kao

$$\mathsf{PRIMES} = \{p \mid p \text{ je prim broj}\}.$$

Lako je vidjeti da je $\mathsf{PRIMES} \in \mathsf{coNP}$ jer je certifikat npr. jedan od djeljitelja zadanog broja. Dok je puno teže pronaći certifikat primarnosti nekog broja.

3.1. PRIMES je u klasi NP

Certifikat primarnosti nekog broja koji bi PRIMES svrstao u $\mathsf{NP} \cap \mathsf{coNP}$ je 1975. našao Vaughan R. Pratt (Pratt, 1975). To je bio značajan rezultat i dugo vremena je PRIMES bio primjer problema koji se nalazi u $\mathsf{NP} \cap \mathsf{coNP}$ a za koji nije bilo poznato je li u klasi P.

Prattov dokaz se temeljni na činjenici da je $\mathbb{Z}_p^*$ ciklička grupa reda $p-1$ ako i samo ako je p prim broj. To možemo zapisati i na sljedeći način bez spominjanja grupa kao što je viđeno u (Papadimitriou, 1995).

Teorem. *Broj $p > 1$ je prost ako i samo ako postoji broj $1 < r < p$ takav da je $r^{p-1} \equiv 1 \pmod{p}$ i vrijedi $r^{\frac{p-1}{q}} \not\equiv 1 \pmod{p}$ za sve proste djeljitelje q broja $p-1$.*

Vidimo da je ovdje r u biti generator grupe $\mathbb{Z}_p^*$. Certifikat se onda sastoji od nekog r zajedno sa svim q-ovima. Dodatno, potrebno je paziti da su svi q-ovi prim brojevi pa se i za svakog od njih priloži takav certifikat (rekurzivno). Zatim je potrebno dokazati da je takav certifikat ispravan (što je posljedica Teorema 3.1), da ga je moguće provjeriti u polinomnom vremenu i da je duljine polinomno ograničene duljinom broja čiju primarnost pokazuje.

3.2. Daljni rezultati o položaju PRIMES

Ubrzo nakon Pratta, 1976. je Gary L. Miller pronašao deterministički polinomni algoritam za PRIMES ali pod uvjetom Proširene Riemannove hipoteze! Kasnije je Michael O. Rabin izmjenio Millerov algoritam u algoritam sa slučajnim elementima (engl. *randomized algorithm*) koji više nije ovisio o Proširenoj Riemannovoj hipotezi.

Važni rezultati koji su učvrstili položaj PRIMES među vjerojatnosnim klasama složenosti su 1976. Roberta M. Solovaya i Volkera Strassena, te 1986. Shafia Goldwassera i Joea Kiliana.

Leonard Adleman i Carl Pomerance su 1983. dali algoritam za PRIMES složenosti $(\log n)^{\mathcal{O}(\log\log\log n)}$ za traženi broj n što još uvijek nije unutar klase P.

U kolovozu 2002. Manindra Agrawal, profesor na Indijskom institutu tehnologije Kanpur (IIT Kanpur), zajedno s dvojicom studenata pod njegovim mentorstvom, Neerayom Kayalom i Nitinom Saxenom, objavljuju rad (Agrawal et al., 2004) u kojem su predstavili postupak koji u polinomnom vremenu ispituje primarnost, danas poznat kao AKS algoritam.

3.3. Osnovna ideja algoritma

Ideju na kojoj se temelji AKS algoritam možemo uočiti na Pascalovom trokutu!

$$
\begin{array}{ccccccccccc}
 & & & & & 1 & & & & & \\
 & & & & 1 & & 1 & & & & \\
 & & & \boxed{1} & & \boxed{2} & & \boxed{1} & & & \\
 & & & \boxed{1} & & \boxed{3} & & \boxed{3} & & \boxed{1} & \\
 & & & 1 & & 4 & & 6 & & 4 & & 1 \\
 & & \boxed{1} & & \boxed{5} & & \boxed{10} & & \boxed{10} & & \boxed{5} & & \boxed{1} \\
 & 1 & & 6 & & 15 & & 20 & & 15 & & 6 & & 1 \\
\boxed{1} & & \boxed{7} & & \boxed{21} & & \boxed{35} & & \boxed{35} & & \boxed{21} & & \boxed{7} & & \boxed{1} \\
1 & & 8 & & 28 & & 56 & & 70 & & 56 & & 28 & & 8 & & 1
\end{array}
$$

Slika 3.1: Pascalov trokut

Uokvireni brojevi pripadaju redovima s prim rednim brojem (počinje od nule) odnosno odgovaraju binomnom koeficijentu $\binom{p}{i}, 0 \leq i \leq p$. Primijetimo da su

brojevi u sredini uokvirenih redova višekratnici tog prim broja, dok za ostale redove to ne vrijedi. Iskazat ćemo to svojstvo sljedećim teoremom.

Teorem 3.1. *Neka je $n > 1$ cijeli broj, n je prim broj ako i samo ako za sve $a \in \mathbb{Z}_n$, $nzd(a, n) = 1$ vrijedi sljedeća jednakost u prstenu $\mathbb{Z}_n[X]$:*

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

Dokaz.

$$(X + a)^n = X^n + a^n + \sum_{i=1}^{n-1} \binom{n}{i} a^i X^{n-i}$$

Pretpostvimo da je n prim broj, onda je jednakost posljedica Leme 2.1 jer su svi koeficijenti $\binom{n}{i}$ jednaki nuli u $\mathbb{Z}_n$. Također je $a^n \equiv a \pmod{n}$ zbog Fermatovog malog teorema (Teorem 2.2).

Pokažimo drugi smjer — ako je n složen onda jednakost ne vrijedi. Uzmimo jedan od prim faktora broja n i označimo ga s q. Neka se q pojavljuje *točno* k puta u n tj. $q^k \mid n$, i $q^{k+1} \nmid n$. Cilj nam je pokazati da q^k (a time ni n) ne dijeli $\binom{n}{q}$. Po pretpostavci teorema n i a su relativno prosti pa n ne dijeli a^q. Odnosno, koeficijent kraj X^{n-q} neće biti nula!

Promotrimo binomni koeficijent kraj X^{n-q},

$$\binom{n}{q} = \frac{n(n-1)\dots(n-q+1)}{q!}.$$

U brojniku je jedino n djeljiv s q stoga se q u brojniku nalazi točno k puta. U nazivniku se q nalazi točno jednom. Zbog toga q^{k-1} dijeli $\binom{n}{q}$ ali ga ne dijeli q^k. □

Sada bi za broj n mogli izračunati koeficijente polinoma $(X + a)^n$ iz $\mathbb{Z}_n[X]$ i provjeriti njihovu djeljivost s n. Po prethodnom teoremu bi na temelju toga mogli odrediti je li broj n složen ili prost. Takav algoritam i dalje nije efikasan jer je potrebno provjeriti n koeficijenata, a znamo da n eksponencijalno ovisi o broju bitova s kojima je zapisan.

Ono ključno što su primijetili Agrawal, Kayal i Saxena je da možemo ispitivati

$$(X + a)^n \equiv X^n + a \pmod{X^r - 1}$$

kako bi provjerili je li n prim broj. Potrebno je izabrati dovoljno mali r i nekoliko a-ova. Očito je da r kao i broj a-ova mora biti polinomno ograničen s obzirom na $\log n$.

3.4. AKS algoritam

Ulaz: prirodni broj $n > 1$.

1. Ako je $n = a^b$ za neki $a \in \mathbb{N}$ i $b > 1$: vrati `COMPOSITE`
2. Nađi najmanji r takav da je $o_r(n) > \log^2 n$
3. Ako je $\text{nzd}\,(a, n) \neq 1$ za neki $a \leq r$: vrati `COMPOSITE`
4. Ako je $n \leq r$: vrati `PRIME`
5. Za sve $a = 1$ do $\left\lfloor \sqrt{\phi(r)} \log n \right\rfloor$:
   Ako u $\mathbb{Z}_n[X]$ je $(X + a)^n \not\equiv X^n + a \pmod{X^r - 1}$: vrati `COMPOSITE`
6. Vrati `PRIME`

U prvom koraku obavimo provjeru je li n, broj čiju primarnost ispitujemo, savršena potencija i taj korak pokriva neke posebne slučajeve za koje ostatak algoritma ne bi radio. Iz sličnog razloga četvrti korak je bitan samo za $n \leq 5690034$ tj. kasnije ćemo vidjeti da je $r = \mathcal{O}\left(\log^5 n\right)$.

3.5. Točnost algoritma

Teorem 3.2. *AKS algoritam vraća* `PRIME` *ako i samo ako je n prim broj.*

Dokaz teorema izložit ćemo u nastavku. Jedan smjer je jednostavan.

Lema 3.1. *Ako je n prim broj onda AKS algoritam vraća* `PRIME`*.*

Dokaz. Ova lema je posljedica Teorema 3.1. Zbog pretpostavke da je n prim broj algoritam prolazi kroz 1. i 3. korak bez vraćanja `COMPOSITE`. Petlja u 5. koraku također ne vraća `COMPOSITE` jer su sve jednakosti zadovoljene (Teorem 3.1). Dakle, algoritam vraća `PRIME` u 6. ili 4. koraku. □

Lema 3.2. *Ako algoritam vrati* `PRIME` *onda je n prim broj.*

Drugi smjer dokaza iskazan prethodnom lemom je značajno teži pa ćemo se prvo pripremiti za njega. Uvest ćemo neke definicije, oznake i opažanja te učvrstiti neke brojeve za ostatak ovog odjeljka:

- n je broj na ulazu i nije oblika a^b za neki $a \in \mathbb{N}$ i $b > 1$ jer bi inače algoritam vratio `COMPOSITE` u 1. koraku

- r je broj iz 2. koraka algoritma takav da je n reda većeg od $\log^2 n$ u $\mathbb{Z}_r^*$
- neka je p jedan od prostih djeljitelja broja n kojemu je red veći od 1 u $\mathbb{Z}_r^*$ (takav postoji jer je n reda većeg od 1 u $\mathbb{Z}_r^*$). Vrijedi i $p > r$ jer bi inače algoritam ispravno ispitao primarnost u 3. ili 4. koraku
- $n, p \in \mathbb{Z}_r^*$ jer je $\text{nzd}\,(n, r) = \text{nzd}\,(p, r) = 1$ (opet zbog 3. i 4. koraka)
- označimo s $l = \left\lfloor \sqrt{\phi(r)} \log n \right\rfloor$, broj jednakosti koje ćemo ispitati u 5. koraku

Time smo učvrstili brojeve n, p, r i l. U 5. koraku algoritam ispituje l jednakosti koje moraju biti ispunjene zbog pretpostavke da algoritam ne vraća `COMPOSITE`. To su sljedeće jednakosti u prstenu $\mathbb{Z}_n[X]$:

$$(X + a)^n \equiv X^n + a \pmod{X^r - 1}, \quad 0 \leq a \leq l. \tag{1}$$

Kako $p \mid n$ onda te jednakosti vrijede i u prstenu $\mathbb{Z}_p[X]$ što je u suštini posljedica svojstva kongruencije (vidjeti Propoziciju 2.1, točka 2). Također zbog Teorema 3.1 u prstenu $\mathbb{Z}_p[X]$ vrijedi

$$(X + a)^p \equiv X^p + a \pmod{X^r - 1}, \quad 0 \leq a \leq l. \tag{2}$$

Iz jednakosti (1) i (2) u prstenu $\mathbb{Z}_p[X]$ vrijedi i:

$$(X + a)^{n/p} \equiv X^{n/p} + a \pmod{X^r - 1}, \quad 0 \leq a \leq l. \tag{3}$$

Sada ćemo uvesti jedno općenito svojstvo nekih polinoma.

Definicija 3.2. Za polinom $f(X) \in \mathbb{Z}_p[X]$ i broj $m \in \mathbb{N}$ kažemo da je m introspektivan za $f(X)$ ako vrijedi:

$$(f(X))^m \equiv f(X^m) \pmod{X^r - 1}.$$

Iz (2) i (3) slijedi da su n i n/p introspektivni za $X + a$ kad je $0 \leq a \leq l$. A iz definicije i Teorema 3.1 znamo da je očito i p introspektivan za $X + a$ kad je $0 \leq a \leq l$.

Lema 3.3. *Ako su m i m' introspektivni za $f(X) \in \mathbb{Z}_p[X]$ onda je i $m \cdot m'$.*

Dokaz. Po pretpostavci je m introspektivan za $f(X)$ pa imamo:

$$\begin{aligned}(f(X))^{m \cdot m'} &\equiv (f(X^m))^{m'} \pmod{X^r - 1} \\ &\overset{(1)}{\equiv} f(X^{m \cdot m'}) \pmod{X^{m \cdot r} - 1} \\ &\overset{(2)}{\equiv} f(X^{m \cdot m'}) \pmod{X^r - 1}\end{aligned}$$

Jednakost (1) slijedi iz Defincije 3.2 kad supstituiramo m s m' i X s X^m. Jednakost (2) vrijedi jer X^r-1 dijeli $X^{m\cdot r}-1$ u što se možemo uvjeriti ako promatramo sumu prvih n članova geometrijskog niza:

$$1+q+\ldots+q^n=\frac{q^{n+1}-1}{q-1},\ \text{tj.}$$

$$(q-1)(1+q+\ldots+q^n)=q^{n+1}-1$$

i zatim supstituiramo $q=X^r$ i $n+1=m$:

$$X^{mr}-1=(X^r-1)(1+X^r+\ldots+X^{r(m-1)}).$$

Sad je jasno da je X^r-1 jedan od faktora polinoma $X^{mr}-1$. □

Lema 3.4. *Ako je m introspektivan za $f(X), g(X)\in\mathbb{Z}_p[X]$ onda je introspektivan i za $f(X)\cdot g(X)$.*

Dokaz. Po definiciji imamo:

$$\begin{aligned}(f(X)\cdot g(X))^m &\equiv (f(X))^m\cdot(g(X))^m \pmod{X^r-1}\\ &\equiv f(X^m)\cdot g(X^m) \pmod{X^r-1}\end{aligned}$$

□

Definiramo skup $I=\{(\frac{n}{p})^i p^j \mid i,j\geq 0\}$ čiji su svi elementi introspektivni za sve polinome iz skupa $P=\{\prod_{a=0}^{l}(X+a)^k \mid k\geq 0\}$ što je izravna posljedica prethodne dvije leme.

Nad skupom I definiramo prvu grupu koja će biti ključna za dokaz. Grupu G definiramo nad skupom I modulo r. Svi elementi iz I su relativno prosti s r pa je G podgrupa od $\mathbb{Z}_r^*$. Grupu G generiraju n i p i kako je G podgrupa od $\mathbb{Z}_r^*$, a vrijedi $o_r(n)>\log^2 n$ onda je i $|G|>\log^2 n$. Red grupe G ćemo označiti s $|G|=t$.

Definicija 3.3. Definiramo n-ti ciklotonički polinom $\Phi_n(X)$ nad poljem F za $n\in\mathbb{N}$ kao

$$\Phi_n(X)=\prod_{\omega}(X-\omega),$$

umnožak je nad svim n-tim primitivnim korijenima jedinice [1] ω u polju F.

[1] $\omega^n=1$ i n je najmanji takav broj

Primjer 3.1. *Pogledajmo neke $\Phi_n(X)$ u prstenu $\mathbb{Z}_5[X]$:*

$$\Phi_1(X) = X - 1$$
$$\Phi_2(X) = X - 4$$
$$\Phi_4(X) = (X-2)(X-3) = X^2 + 1$$

Iskazat ćemo jedno važno svojstvo koje nećemo dokazivati.

Lema 3.5. *Za sve $n \in \mathbb{N}$ vrijedi jednakost*

$$\prod_{d|n} \Phi_d(X) = X^n - 1.$$

Dakle, vidimo da ako d dijeli n onda i $\Phi_d(X)$ dijeli $X^n - 1$.

Primjer 3.2. *Izračunajmo umnožak $\prod_{d|2} \Phi_d(X)$ u prstenu $\mathbb{Z}_5[X]$:*

$$\prod_{d|2} \Phi_d(X) = \Phi_1(X)\Phi_2(X) = (X-1)(X-4) = X^2 + 4 = X^2 - 1.$$

Sa F_p označavamo konačno polje s p elemenata gdje je p prim broj.

Lema 3.6. *Ako je p prim broj i $h(X)$ ireducibilan polinom stupnja d u F_p onda je $F_p[X]$ modulo $h(X)$ konačno polje reda p^d.*

Neka je $\Phi_r(X)$ r-ti ciklotonički polinom nad F_p onda se $\Phi_r(X)$ može rastaviti na ireducibilne faktore stupnja $o_r(p)$ (ovo nećemo dokazivati). Sa $h(X)$ ćemo označiti jedan takav faktor. On je stupnja većeg od jedan jer je $o_r(p) > 1$.

Definirat ćemo drugu važnu grupu. To je grupa koja se sastoji od svih polinoma iz P nad $\mathbb{Z}_p$, modulo $h(X)$. Označit ćemo je sa $\mathcal{G}$. To je grupa generirana s $X, X+1, \ldots, X+l$ i čini podgrupu multiplikativne grupe polja $F_p[X]$ modulo $h(X)$.

Cilj nam je procijeniti donju i gornju granicu za $|\mathcal{G}|$.

Lema 3.7. $|\mathcal{G}| \geq \binom{t+l}{t-1}$.

Dokaz. Polinom $h(X)$ je jedan od faktora ciklotoničkog polinoma $\Phi_r(X)$ pa po Lemi 3.5 dijeli $X^r - 1$ tj. u $F_p[X]$ imamo:

$$X^r \equiv 1 \pmod{h(X)},$$

štoviše, X je r-ti primitivni korijen jedinice u $F_p[X]$ modulo $h(X)$.

Sad nam je cilj pokazati da bilo koja dva različita polinoma stupnja manjeg od $t = |G|$ iz P će se preslikati u dva različita polinoma unutar $\mathcal{G}$. Neka su $f(X)$ i $g(X)$ takva dva polinoma iz P. Pretpostavimo $f(X) = g(X)$ u polju $F_p[X]$ modulo $h(X)$ i pokažimo kontradikciju.

Neka je $m \in I$, znamo da je on sigurno introspektivan za $f(X)$ i $g(X)$ (u ovom slučaju nemamo modulo $X^r - 1$ no $h(X) \mid X^r - 1$ pa i dalje vrijedi introspekcija), imamo u $F_p[X]$:

$$\begin{aligned} f(X) &= g(X) \pmod{h(X)} \\ (f(X))^m &= (g(X))^m \pmod{h(X)} \\ f(X^m) &= g(X^m) \pmod{h(X)} \end{aligned}$$

Stoga, za sve $m \in G$, X^m je korijen polinoma $Q(X) = f(X) - g(X)$ u $F_p[X]$ modulo $h(X)$. I ti korijeni su različiti pa je $Q(X)$ stupnja jednakog $t = |G|$, no po izboru $f(X)$ i $g(X)$ koji su stupnja manjeg od t to nije moguće tj. došli smo do kontradikcije i vrijedi $f(X) \neq g(X)$ u polju $F_p[X]$ modulo $h(X)$.

Možemo vidjeti da su svi polinomi $X + a$, $0 \leq a \leq l$ različiti u $F_p[X]$ jer je $l < r$, a znamo da je $p > r$. Također su različiti i u modulo $h(X)$ jer je stupanj od $h(X)$ veći od jedan. Stoga imamo bar $l + 1$ različit polinom stupnja jedan u $\mathcal{G}$.

Iz ta dva opažanja slijedi da imamo bar $\binom{t+l}{t-1}$ različitih polinoma stupnja manjeg od t u grupi $\mathcal{G}$. □

Sad ćemo procijeniti i donju granicu za $\mathcal{G}$.

Lema 3.8. *Ako n nije potencija broja p onda $|\mathcal{G}| \leq n^{\sqrt{t}}$.*

Dokaz. Konstruirat ćemo jedan podskup od I:

$$I' = \{(n/p)^i p^j \mid 0 \leq i, j \leq \lfloor \sqrt{t} \rfloor\}.$$

Po pretpostavci da n nije potencija broja p imamo da skup I' sadrži bar $(\lfloor \sqrt{t} \rfloor + 1)^2 > t$ različitih elemenata. Stoga bar dva elementa iz I' moraju biti jednaka u modulo r jer je $|G| = t$. Označimo neka takva dva broja s m_1 i m_2 i pretpostavimo $m_1 > m_2$. Onda imamo:

$$X^{m_1} \equiv X^{m_2} \pmod{X^r - 1}.$$

Neka je sad $f(X) \in P$ pa u $F_p[X]$ imamo:

$$\begin{aligned}(f(X))^{m_1} &\equiv f(X^{m_1}) \pmod{X^r - 1} \\ &\equiv f(X^{m_2}) \pmod{X^r - 1} \\ &\equiv (f(X))^{m_2} \pmod{X^r - 1}.\end{aligned}$$

Dakle, $(f(X))^{m_1} = (f(X))^{m_2}$ u polju $F_p[X]$ modulo $h(X)$ (jer $h(X)$ dijeli $X^r - 1$) pa je $f(X)$ korijen polinoma $Q'(Y) = Y^{m_1} - Y^{m_2}$. Polinom $Q'(X)$ ima bar $|\mathcal{G}|$ različitih korijena u polju $F_p[X]$ modulo $h(X)$ jer je izbor $f(X) \in \mathcal{G}$ proizvoljan. Ali znamo i gornju granicu za stupanj polinoma $Q'(X)$, ona je:

$$m_1 \leq \left(\frac{n}{p} \cdot p\right)^{\lfloor\sqrt{t}\rfloor} \leq n^{\sqrt{t}}.$$

Iz toga slijedi granica za red grupe $\mathcal{G}$, $|\mathcal{G}| \leq n^{\sqrt{t}}$.

□

Sad možemo dokazati Lemu 3.2:

Dokaz. Po Lemi 3.7 imamo nejednakost:

$$\begin{aligned}|\mathcal{G}| &\geq \binom{t+l}{t-1} \\ &\overset{(1)}{\geq} \binom{l+1+\lfloor\sqrt{t}\log n\rfloor}{\lfloor\sqrt{t}\log n\rfloor} \\ &\overset{(2)}{\geq} \binom{1+2\lfloor\sqrt{t}\log n\rfloor}{\lfloor\sqrt{t}\log n\rfloor} \\ &\overset{(3)}{>} 2^{\lfloor\sqrt{t}\log n\rfloor+1} \\ &\geq n^{\sqrt{t}}.\end{aligned}$$

Nejednakost (1) vrijedi jer je $t > \sqrt{t}\log n$, nejednakost (2) jer je $l = \lfloor\sqrt{\phi(r)}\log n\rfloor \geq \lfloor\sqrt{t}\log n\rfloor$, i nejednakost (3) jer je $\lfloor\sqrt{t}\log n\rfloor > \lfloor\log^2 n\rfloor \geq 1$.

No po Lemi 3.8: $|\mathcal{G}| \leq n^{\sqrt{t}}$ ako n nije potencija broja p — dakle, n je potencija broja p no ako nije $n = p$ onda bi algoritam vratio `COMPOSITE` u 1. koraku što bi bila kontradikcija. Prema tome, $n = p$. □

Time je dokazan i Teorem 3.2.

3.6. Vremenska složenost algoritma

Prvo ćemo razmotriti složenosti nekih operacija koje se koriste u AKS algoritmu. Granice složenosti koje ćemo dati bit će vrlo grube jer nam je cilj pokazati samo polinomnu složenost AKS algoritma. Nećemo ulaziti ni u formalnosti već ćemo samo dati ideje za te algoritme.

Počnimo od osnovnih aritmetičkih operacija. Zbrajanje (i oduzimanje) dva broja a i b (pretpostavimo $a \geq b$) je moguće učiniti u vremenu $\mathcal{O}(\log a)$ jer prolazimo redom po bitovima zbrajajući ih. Množenje (i dijeljenje) brojeva a i b je moguće učiniti "školskom metodom" i složenost toga je $\mathcal{O}\left(\log^2 a\right)$.

U $\mathbb{Z}_n$ možemo obavljati iste operacije tako da računamo s reprezentantima manjim od n, a ukoliko rezultat operacije bude veći od n onda ga jednostavno umanjimo za n.

Za potenciranje a^k u $\mathbb{Z}_n$ nije potrebno učiniti k množenja već možemo potencirati broj uzastopnim kvadriranjem. Ideja za algoritam potječe od sljedeće rekurzivne relacije za $\text{pow}(a,k) = a^k$:

$$\text{pow}(a,k) = \begin{cases} 1 & \text{ako je } k = 0 \\ \text{pow}^2(a, k/2) & \text{ako je } k \text{ paran} \\ a \cdot \text{pow}(a, k-1) & \text{ako je } k \text{ neparan} \end{cases}$$

Teorem 3.3. *Vremenska složenost algoritma za potenciranje brojeva $a \in \mathbb{Z}_n$ i $k \in \mathbb{N}$, a^k je $\mathcal{O}\left(\log(k)\log^2(a)\right)$.*

Najveći zajednički djeljitelj brojeva a i b koji zapisujemo $\text{nzd}(a,b)$ možemo pronaći pomoću poznatog Euklidovog algoritma. Neka je $a = q \cdot b + r$. Vrijedi da je

$$\text{nzd}(a,b) = \text{nzd}(qb + r, b) = \text{nzd}(r,b) = \text{nzd}(b,r). \qquad (*)$$

Druga jednakost slijedi iz toga što je qb djeljivo s b. Zapažanje takve rekurzivne veze nam daje osnovu za algoritam. Očito je $\text{nzd}(r,0) = r$ pa znamo i uvjet za zaustavljanje algoritama.

Teorem 3.4. *Vremenska složenost Euklidovog algoritma za traženje najvećeg zajedničkog djeljitelja nzd(a,b), $a \geq b$ je $\mathcal{O}\left(\log^3 a\right)$.*

Dokaz. Pitamo se koliko puta moramo ponoviti rekurzivni korak u (*) da bi došli do rješenja. Promotrimo jedan od njih:

$$\text{nzd}(r_{i-1}, r_i) = \text{nzd}(r_i, r_{i+1}) \text{ gdje je } r_{i-1} = kr_i + r_{i+1}$$

odnosno r_{i+1} je ostatak dijeljenja r_{i-1} s r_i pa vrijedi:

$$r_{i-1} \geq r_i + r_{i+1} > 2r_{i+1}$$

Pomnožimo cijelu nejednakost s r_i i malo posložimo:

$$r_{i+1}r_i < \frac{r_i r_{i-1}}{2}$$

Odnosno, ako to proširimo do r_0 i r_1 (tj. a i b čiji najveći zajednički djeljitelj tražimo):

$$r_{i+1}r_i < \frac{r_i r_{i-1}}{2} < \frac{r_1 r_0}{2^i}$$

Ako je $r_{i+1}r_i < 1$ onda će algoritam stati jer je sigurno jedan od njih 0. Sad lako dobijemo za koje i vrijedi ta nejednakost:

$$r_{i+1}r_i < \frac{r_1 r_0}{2^i} < 1$$

$$r_1 r_0 < 2^i$$

$$i > \log_2(r_1 r_0)$$

Dakle, nakon $\mathcal{O}(\log(r_1 r_0))$ tj. $\mathcal{O}(\log(ab)) = \mathcal{O}(\log(a))$ koraka algoritam će sigurno stati.

U svakom koraku moramo obaviti dijeljenje što je složenosti $\mathcal{O}(\log(a)\log(b)) = \mathcal{O}\left(\log^2(a)\right)$ pa je konačna složenost algoritma $\mathcal{O}\left(\log^3(a)\right)$. □

Granica se može poboljšati na $\mathcal{O}(\log(a)\log(b))$ jer u svakom koraku dijelimo sve manje i manje brojeve.

Teorem 3.5. *Određivanje k-tog korijena je problem u kojem za zadane $a \in \mathbb{N}$ i $k \in \mathbb{N}$ moramo odrediti $\lfloor \sqrt[k]{a} \rfloor$. Postoji algoritam za određivanje k-tog korijena vremenske složenosti $\mathcal{O}\left(\log(k)\log^3(a)\right)$.*

Ideja je da binarnim pretraživanjem po $a' = 1 \ldots a$ ispitujemo potenciranjem je li $(a')^k = a$ (u biti moramo biti malo pažljiviji jer k-ti korijen ne mora biti cijeli broj). Traženje će trajati $\mathcal{O}(\log a)$ koraka i svaki put moramo izvršiti potenciranje složenosti $\mathcal{O}\left(\log k \log^2 a\right)$.

Teorem 3.6. *Ispitivanje je li zadani broj $n \in \mathbb{N}$ k-ta potencija je problem u kojem moramo provjeriti postoji li $a \in \mathbb{N}$ takav da je $a^k = n$. Postoji algoritam za ispitivanje k-te potencije vremenske složenosti $\mathcal{O}\left(\log(k)\log^3(n)\right)$.*

Možemo modficirati algoritam za određivanje k-tog korijena tako da izračunamo $\lfloor \sqrt[k]{n} \rfloor$ i provjerimo je li $\lfloor \sqrt[k]{n} \rfloor^k = n$.

Teorem 3.7. *Ispitivanje je li zadani broj $n \in \mathbb{N}$ savršena potencija je problem u kojem moramo provjeriti postoje li $a \in \mathbb{N}$ i $k \in \mathbb{N}, k > 1$ takvi da je $a^k = n$. Postoji algoritam za ispitivanje savršene potencije vremenske složenosti $\mathcal{O}\left(\log^4(n) \log\log n\right)$.*

Najmanji a (ako je $n > 1$) je 2 pa je najveći mogući $k = \log_2 n$. Ideja je da pretražujemo po $k' = 2 \dots \log_2 n$ uzastopce i ispitujemo je li n k'-ta potencija. Pretraživanje će trajati $\mathcal{O}(\log n)$ koraka i svaki put moramo ispitati je li n k'-ta potencija što je složenosti $\mathcal{O}\left(\log(k') \log^3(n)\right) = \mathcal{O}\left(\log\log(n) \log^3(n)\right)$ tj. ukupna složenost je $\mathcal{O}\left(\log^4(n) \log\log n\right)$.

Dolazimo do važnog dijela za dokaz složenosti AKS algoritma. U 2. koraku algoritma tražimo r takav da je $o_r(n) > \log^2 n$ (n je broj na ulazu) i bitno je da naiđemo na r dovoljno brzo tj. u vremenu polinomnom za n. Sljedećim teoremom ćemo se uvjeriti u to, no prvo će nam biti potrebna jedna lema o gustoći distribucije prim brojeva.

Lema 3.9. *Postoje $c_1, c_2, x_0 > 0$ takvi da za sve $x > x_0$:*

$$c_1 x \leq \sum_{p \leq x} \log(p) \leq c_2 x,$$

gdje je suma nad svim prim brojevima p manjim ili jednakim x.

Funkcija koja x-u pridružuje $\sum_{p \leq x} \log(p)$ je u literaturi poznata kao Čebiševljeva theta funkcija i u asimptotskoj notaciji se jednostavno zapisuje $\sum_{p \leq x} \log(p) = \Theta(x)$.

Teorem 3.8. *Za cijele brojeve $n > 1$ i $m \geq 1$, najmanji prim broj r takav da $r \nmid n$ i $o_r(n) > m$ je $\mathcal{O}(m^2 \log n)$.*

Dokaz. Reći ćemo da je r nezadovoljavajući ako vrijedi da ili $r \mid n$ ili $r \mid (n^d - 1)$ za neki $d = 1, 2 \dots m$ (jer je u tom slučaju $o_r(n) \leq m$). Dakle, za svaki nezadovoljavajući r vrijedi:

$$r \mid n \prod_{d=1}^{m} (n^d - 1).$$

Ako su svi prim brojevi r do neke zadane granice $x \geq 2$ nezadovoljavajući onda umnožak svih tih r-ova dijeli $\prod_{d=1}^{m}(n^d - 1)$ pa imamo:

$$\prod_{r \leq x} r \leq \prod_{d=1}^{m} (n^d - 1).$$

Uzmemo li logaritme s obje strane dobivamo

$$\sum_{r\leq x}\log(r) \leq \log\left(\prod_{d=1}^{m}(n^d-1)\right) \leq (\log n)\left(1+\sum_{d=1}^{m} d\right)$$
$$= (\log n)(1+\frac{m(m+1)}{2}).$$

Ali po Lemi 3.9 imamo za neku konstantu $c > 0$:

$$\sum_{r\leq x}\log(r) \geq cx.$$

Iz toga slijedi da je

$$x \leq \frac{1}{c}(\log n)(1+\frac{m(m+1)}{2}),$$

a po pretpostavci da je x gornja granica za nezadovoljavajuće r-ove slijedi da će prvi prim broj veći od x zadovoljiti uvjete teorema i da je taj $r = \mathcal{O}(m^2 \log n)$. □

U 2. koraku AKS algoritma je $m = \log^2 n$ pa ćemo uzastopnim ispitivanjem r-ova doći do traženog u najviše $\mathcal{O}\left(\log^5 n\right)$ koraka.

Teorem 3.9. *Vremenska složenost AKS algoritma je* $\mathcal{O}\left(\log^{16.5} n\right)$.

Dokaz. Neka je na ulazu broj n. Prvi korak algoritma možemo obaviti u vremenu $\mathcal{O}\left(\log^4(n)\log\log(n)\right)$ što je posljedica Teorema 3.7.

U drugom koraku tražimo r takav da je $o_r(n) > \log^2 n$. To možemo učiniti tako da promatramo uzastopne vrijednosti od r i provjeravamo je li $n^k \not\equiv 1 \pmod r$ za sve $k \leq \log^2 n$. Po Teoremu 3.8 znamo da ćemo na traženi r naići u $\mathcal{O}\left(\log^5 n\right)$ koraka, a u svakom koraku moramo napraviti $\log^2 n$ potenciranja što je složenosti $\mathcal{O}\left(\log^2(n)\log^2(n)\log\log^2(n)\right) = \mathcal{O}\left(\log^4(n)\log\log^2(n)\right)$ (Teorem 3.3), odnosno, ukupna složenost drugog koraka je $\mathcal{O}\left(\log^9(n)\log\log^2(n)\right)$.

Treći korak zahtjeva izračunavanje najvećeg zajedničkog djeljitelja $\text{nzd}(a, n)$ za sve $a \leq r = \mathcal{O}\left(\log^5 n\right)$. Iz Teorema 3.4 za složenost Euklidovog algortima slijedi da je složenost trećeg koraka $\mathcal{O}\left(\log^8 n\right)$.

U četvrtom koraku je potrebno učiniti jedno uspoređivanje i to je složenosti $\mathcal{O}(\log n)$.

Peti korak traži da provjerimo $\left\lfloor\sqrt{\phi(r)}\log n\right\rfloor$ jednadžbi. U svakoj jednadžbi s lijeve strane imamo $(X+a)^n$ što zahtjeva $\log n$ množenja polinoma stupnja $r = \mathcal{O}\left(\log^5 n\right)$ (zbog modulo $X^r - 1$) s koeficijentima iz $\mathbb{Z}_n$ (dakle, duljine $\mathcal{O}(\log n)$). Svako množenje polinoma zahtjeva $\mathcal{O}(r^2)$ množenja složenosti $\mathcal{O}\left(\log^2 n\right)$. Stoga je ukupna složenost petog koraka

$$\mathcal{O}\left(r^2\sqrt{\phi(r)}\log^4 n\right) = \mathcal{O}\left(r^{5/2}\log^4 n\right) = \mathcal{O}\left(\log^{33/2} n\right).$$

Očito je da peti korak dominara nad ostalima po složenosti pa je ukupna složenost AKS algoritma $\mathcal{O}\left(\log^{16.5} n\right)$. □

Time smo pokazali da algoritam AKS radi u polinomnom vremenu. Odnosno, uz Teorem 3.2 imamo $\mathsf{PRIMES} \in \mathsf{P}$.

4. Zaključak

Znamo da teorija brojeva i složenost imaju veliku ulogu u današnjoj kriptografiji i koliko nam je ona bitna u modernom svijetu. Bitno nam je da možemo lako generirati i ispitati prim brojeve i pouzdajemo se da nije moguće efikasno faktorizirati velike brojeve pomoću klasičnog računala iako za ovaj problem nemamo jasno utvrđenu klasu složenosti kao što imamo za PRIMES.

Rad Agrawala, Kayala i Saxene ima veliki značaj za teoriju ali u praksi se i dalje koriste druge, starije metode poput Miller-Rabinovog testa. Osim što je prvi neuvjetni dokaz da je $\mathsf{PRIMES} \in \mathsf{P}$ značajan je i po svojoj jednostavnosti i elementarnosti pa je i zanimljivo kako je sve do današnjih dana izbjegavao otkriću.

Literatura

Manindra Agrawal, Neeral Kayal, i Nitin Saxena. Primes in P. *Annals of Mathematics*, 160(2):781–793, 2004.

Folkmar Bornemann. PRIMES is in P: A breakthrough for 'everyman.'. *Notices Amer. Math. Soc.*, 50:545–552, 2003.

Christos H. Papadimitriou. *Computational Complexity*. Addison Wesley, 1995.

Vaughan R. Pratt. Every prime has a succinct certificate. *SIAM Journal on Computing*, 4(3):214–220, 1975.

Victor Shoup. *A Computational Introduction to Number Theory and Algebra*. Cambridge University Press, 2008. URL `http://www.shoup.net/ntb/`.

Michael Sipser. *Introduction to the Theory of Computation, 2. izdanje*. Course Technology, 2005.