Euklidov algoritam

Blagojević, Dragana

Master's thesis / Diplomski rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Applied Mathematics and Informatics / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet primijenjene matematike i informatike**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:126:590387

Sveučilište J. J. Strossmayera u Osijeku
Fakultet primijenjene matematike i informatike
Sveučilišni diplomski studij matematike, smjer: financijska matematika i statistika

Dragana Blagojević

Euklidov algoritam

Diplomski rad

Osijek, 2023.

Sveučilište J. J. Strossmayera u Osijeku
Fakultet primijenjene matematike i informatike
Sveučilišni diplomski studij matematike, smjer: financijska matematika i statistika

Dragana Blagojević

Euklidov algoritam

Diplomski rad

Mentor: prof. dr. sc. Ivan Matić

Osijek, 2023.

Sadržaj

Uvod

Euklid (oko 330. pr. Kr. - 275. pr. Kr.) je najutjecajniji grčki matematičar, a o njegovom životu nema puno informacija. Zbog svoga velikog doprinosa elementarnoj geometriji, Euklid nosi titulu "oca geometrije". On je najpoznatiji po svojoj knjizi *Elementi*, koja se smatra jednim od najutjecajnijih i najznačajnijih djela u povijesti matematike. *Elementi* su prevedeni na većinu svjetskih jezika što ovo djelo čini jednom od najprevođenijih knjiga na svijetu te se sastoji od 13 dijelova. Euklid je u svome najpoznatijem djelu proučio planimetriju, stereometriju, aritmetiku te teoriju brojeva. Od *VII.* do *X.* knjige opisan je iznimno efikasan način pronalaska najvećeg zajedničkog djelitelja dvaju brojeva kojeg nazivamo Euklidov algoritam. Često je otkriće ovog algoritma bilo pripisivano Pitagorejcima. Također je poznato da su u 5. stoljeću kineski i indijski matematičari koristili navedeni algoritam.
Euklidov algoritam ima veliki značaj u matematici zbog svoje brojne teorijske i praktične primjene. Osim korištenja za pronalaženje najvećeg zajedničkog djelitelja dvaju brojeva, također se koristi i kod skraćivanja razlomka do neskrativog oblika. Osim navedenih primjena, Euklidov algoritam koristi se kod rješavanja linearnih diofantskih jednadžbi, razvoja broja u verižni razlomak te kod pronalaska najbolje aproksimacije realnih brojeva. Svoj značaj opravdava time što se koristi kao osnova u dokazima nekih teorema u kriptografiji i teoriji brojeva. Originalni algoritam koristio se samo za prirodne i realne brojeve (geometrijske duljine), no u 19. stoljeću algoritam je generaliziran te postaje primjenjiv za sve tipove brojeva.

Algoritam se najčešće definira kao postupak ili tehnika za rješavanje nekoga zadatka. Algoritam je sačinjen od niza koraka koji su konačni te koji se trebaju obaviti da bi došli do rješenja. Mi ćemo proučavati jedan od najpoznatijih algoritama, a to je upravo Euklidov algoritam. Dotaknut ćemo se i složenosti algoritma te vremena izvršenja. Vrijeme izvršenja predstavlja vremensku složenost te mjeri vrijeme potrebno da se algoritam izvrši.
Proučit ćemo još i poznati Kineski teorem o ostacima koji nam daje rješenje sustava linearnih kongruencija, a osim njega, spomenut ćemo i Fermatov teorem. Vidjet ćemo koje su njihove primjene i interpretirati ih u smislu algoritma.

Na kraju predstavljamo i praktičniju primjenu teorije brojeva, odnosno primjenu u kriptografiji. Algoritamski opisujemo jedan od najpoznatiji kriptosustava s javnim ključem, tzv. RSA kriptosustav. Njegova sigurnost je zasnovana na problemu faktorizacije velikih složenih brojeva oblika $n = pq$, pri čemu su p, q prosti brojevi. Termin veliki brojevi označava brojeve koji se sastoje od stotinjak znamenaka. Pronalazak tako velikih prostih brojeva nije jednostavan. Stoga možemo zaključiti da je traženje velikih prostih brojeva potaknuto kriptografskim zahtjevima, a između ostaloga i ljudskom znatiželjom.

1 Euklidov algoritam

U ovom radu raspravljamo o Euklidovom algoritmu za pronalaženje najvećeg zajedničkog djelitelja. Ispostavilo se da Euklidov algoritam ima mnoštvo dobrih svojstava te da ima daleko širu primjenu od samog pronalaženja najvećeg zajedničkog djelitelja.

1.1 Osnovni pojmovi

Uvest ćemo osnovne pojmove i tvrdnje koji će nam biti potrebni u uvođenju Euklidovog algoritma.

Definicija 1. *Neka su a i b cijeli brojevi te $a \neq 0$. Kažemo da a dijeli b, odnosno da je b djeljiv s a ukoliko postoji cijeli broj c takav da je $b = a \cdot c$. To označavamo s $a \mid b$. Ukoliko takav c ne postoji, kažemo da a ne dijeli b (ili da b nije djeljiv s a) te označavamo s $a \nmid b$. Nadalje, ukoliko a dijeli b, kažemo da je b višekratnik od a te da je a djelitelj od b.*

Sada ćemo dokazati jedan od fundamentalnih teorema u teoriji brojeva, poznatiji pod nazivom *Teorem o dijeljenju s ostatkom.* Ovaj teorem nam je iznimno važan jer prikazuje jedan korak Euklidovog algoritma.

Teorem 1 (vidjeti [2, Teorem 1.1.2.])**.** *Neka su a, b cijeli brojevi, $a > 0$. Tada postoje jedinstveni cijeli brojevi q i r takvi da je $b = p \cdot a + r$, pri čemu vrijedi $0 \leq r < a$.*

Dokaz. Prvo pokažimo kako postoje brojevi q i r. Pogledajmo racionalan broj $\frac{b}{a}$. Neka je q cijeli broj takav da se broj $\frac{b}{a}$ nalazi u poluotvorenom intervalu $[q, q+1\rangle$. Očigledno je da vrijedi $0 \leq \frac{b}{a} - q < 1$.
Neka je $r = b - a \cdot q = a(\frac{b}{a} - q)$. Nadalje, uočimo kako je r cijeli broj koji zadovoljava $b = q \cdot a + r$. Prema prethodnoj nejednakosti slijedi $0 \leq r < a$. Preostaje dokazati jedinstvenost. Dakle, neka su $b_1 = q_1 \cdot a + r_1$ te $b_2 = q_2 \cdot a + r_2$ dva različita rastava broja b, tj. $q_1, q_2, r_1, r_2 \in \mathbb{Z}$ i $0 \leq r_1, r_2 < a$. Nadalje, oduzmimo prethodna dva rastava i dobijamo $a(q_1 - q_2) = r_2 - r_1$. Ako je $q_1 \neq q_2$, tada a dijeli $r_2 - r_1$. Međutim, vrijedi $-a+1 \leq r_2 - r_1 \leq a-1$ te $\mid r_2 - r_1 \mid < a$. Stoga, vrijedi $q_1 = q_2$, stoga vrijedi i $r_1 = r_2$ čime je dokazana jedinstvenost. □

Broj r iz prethodnog teorema naziva se ostatak pri dijeljenju b s a, dok se broj q naziva kvocijent cjelobrojnog dijeljenja. Kažemo da je b djeljiv s a ako i samo ako vrijedi da je $r = 0$.

Primjer 1. *a) Neka su $b = 10$, $a = 2$. Tada je $10 = 2 \cdot 5$, pri čemu je $q = 5$, a $r = 0$ što je bilo i očekivano jer znamo da $2 \mid 10$.*

b) Ako je $b = 29$, a $a = 7$, iz prethodnog teorema slijedi $27 = 4 \cdot 7 + 1$, pri čemu je kvocijent $q = 4$, a ostatak $r = 1$.

Neka nam je dan skup S (konačan ili beskonačan) čiji su neki elementi različiti od 0. Svaki cijeli broj d koji dijeli svaki element danog skupa nazivamo *zajednički djelitelj*. Budući da svaki zajednički djelitelj dijeli svaki element u S, to znači da on dijeli neki $a \in S$ te $|d| \leq |a|$. Stoga zaključujemo da postoji konačno mnogo zajedničkih djelitelja pa među njima postoji najveći.

Definicija 2. *Neka su $b_0, b_1, \ldots, b_n$ cijeli brojevi koji nisu svi jednaki 0. Najveći zajednički djelitelj brojeva $b_0, b_1, \ldots, b_n$ je najveći cijeli broj d za kojeg vrijedi: $d \mid b_i$, za svaki $i \in \{1, 2, \ldots, n\}$ i označavamo ga s $nzd(b_0, b_1, \ldots, b_n)$.*

Primjer 2. *Najveći zajednički djelitelj od brojeva 10 i 15 je $nzd(10, 15) = 5$, dok najveći zajednički djelitelj od 45 i 19 je $nzd(45, 19) = 1$.*

Za brojeve $a_0, a_1, \ldots, a_n$, koji nisu svi jednaki nuli te čiji je najveći zajednički djelitelj jednak jedan, tj. $nzd(a_0, a_1, \ldots, a_n) = 1$ kažemo da su **relativno prosti**.
Sljedeći teorem daje važnu jednakost koja će nam biti neophodna kod dokazivanja Euklidovog algoritma.

Teorem 2 (vidjeti [1, Teorem 2.3.]). *Neka su b i c cijeli brojevi od kojih je barem jedan različit od 0. Tada vrijedi*

$$nzd(b, c) = min\{bx + cy : x, y \in \mathbb{Z}\} \cap \mathbb{N}.$$

Dokaz. Provjerimo da skup $S = \{bx+cy : x, y \in \mathbb{Z}\}$ ima barem jedan pozitivan element. Bez smanjenja općenitosti, možemo pretpostaviti da je $b \neq 0$. Ako je $b > 0$, onda je $b = b \cdot 1 + c \cdot 0$ pozitivan, a ako je $b < 0$, onda je $|b| = b \cdot (-1) + c \cdot 0$ pozitivan.
Neka je $g = nzd(b, c)$ te pretpostavimo da je l najmanji pozitivni član skupa $S = \{bx + cy : x, y \in \mathbb{Z}\}$. Dakle, možemo zaključiti da postoje cijeli brojevi x_0 i y_0 takvi da je $l = bx_0 + cy_0$. Preostaje pokazati da $l \mid b$ i $l \mid c$. Pretpostavimo da npr. $l \nmid b$. Prema Teoremu 1. postoje cijeli brojevi q i r takvi da je $b = lq + r$ i $0 < r < l$. Imamo

$$r = b - lq = b - q(bx_0 + cy_0) = b(1 - qx_0) + c(-qy_0) \in S,$$

što je u kontradikciji s minimalnosti od l. Stoga, $l \mid b$, a na isti način se pokazuje da $l \mid c$. Prema tome vrijedi da je $l \leq g$. Budući da je $g = nzd(b, c)$, tada postoje $\beta, \gamma \in \mathbb{Z}$ takvi da je $b = g\beta$, $c = g\gamma$, pa je $l = bx_0 + cy_0 = g(\beta x_0 + \gamma y_0)$. Odavde slijedi da je $g \leq l$ te je time dokazano da je $g = l$. □

Posljedica prethodnog teorema se često koristi, a naziva se još i **Bezoutov identitet** te glasi ovako: za sve cijele brojeve a i b postoje cijeli brojevi x i y takvi da je

$$a \cdot x + b \cdot y = nzd(a, b) \in \mathbb{N}.$$

Primjer 3. *Neka je $a = 15$ i $b = 10$. Ranije smo već naveli da vrijedi $nzd(15, 10) = 5$. Uočimo sljedeće $15 \cdot 1 + 10 \cdot (1) = 15 - 10 = 5$, dakle pronašli smo cijele brojeve $x = 1$ i $y = 1$ koji ispunjavaju Bezoutov identitet.*

1.2 Osnovni Euklidov algoritam

Razmatramo sljedeći problem: dana su dva nenegativna cijela broja a i b, izračunajte njihov najveći zajednički djelitelj, $nzd(a,b)$. To možemo učiniti koristeći dobro poznati Euklidov algoritam.

Osnovna ideja Euklidovog algoritma je sljedeća. Bez smanjenja općenitosti, možemo pretpostaviti da je $a \geq b \geq 0$. Ako je $b = 0$, onda nemamo što dokazivati budući da je $nzd(a,0) = a$. Nadalje, ako je $b > 0$ možemo izračunati cjelobrojni kvocijent $q := \lfloor a/b \rfloor$ i prisjetimo se da je $r := a \pmod b$, $0 \leq r < b$. Iz jednakosti

$$a = bq + r,$$

lako možemo vidjeti ako cijeli broj d dijeli i b i r, tada dijeli i a, također, ako cijeli broj d dijeli a i b, tada dijeli i r. Stoga, zaključujemo da vrijedi $nzd(a,b) = nzd(b,r)$, pa postupkom dijeljenja pokušavamo smanjiti problem pronalaska $nzd(a,b)$ "manjim" problemom pronalaska $nzd(b,r)$.

Teorem 3 (vidjeti [3, Teorem 4.1.]). *Neka su a, b cijeli brojevi takvi da je $a \geq b \geq 0$. Koristeći dijeljenje s pripadajućim ostatcima, definiramo cijele brojeve $r_0, r_1, \ldots, r_{l+1}$ i $q_1, \ldots, q_l$, pri čemu je $l \geq 0$, tako da*

$$\begin{aligned}
a &= r_0, \\
b &= r_1, \\
r_0 &= r_1 q_1 + r_2 \qquad (0 < r_2 < r_1), \\
&\vdots \\
r_{i-1} &= r_i q_i + r_{i+1} \qquad (0 < r_{i+1} < r_i) \\
&\vdots \\
r_{l-2} &= r_{l-1} q_{l-1} + r_l \qquad (0 < r_l < r_{l-1}) \\
r_{l-1} &= r_l q_l \qquad (r_{l+1} = 0)
\end{aligned}$$

Prema definiciji, $l = 0$ ako je $b = 0$, inače $l > 0$.
Tada vrijedi $r_l = nzd(a,b)$. Nadalje, ako je $b > 0$, tada $l \leq \log b / \log \phi + 1$ pri čemu je $\phi := (1 + \sqrt{5})/2 \approx 1.62$.

Dokaz. Prva tvrdnja vidi se iz toga da za svaki $i = 1, \ldots, l$ imamo $r_{i-1} = r_i q_i + r_{i+1}$, odakle slijedi da je zajednički djelitelj od r_{i-1} i r_i jednak zajedničkom djelitelju od r_i i r_{i+1}, dakle vrijedi $nzd(r_{i-1}, r_i) = nzd(r_i, r_{i+1})$. Odatle slijedi

$$nzd(a,b) = nzd(r_0, r_1) = nzd(r_l, r_{l+1}) = r_l.$$

Kako bismo dokazali drugu tvrdnju, pretpostavimo da je $b > 0$, dakle i $l > 0$. Ako je $l = 1$, pretpostavka očigledno vrijedi, stoga neka je $l > 1$. Tvrdimo da za $i = 0, 1, \ldots, l-1$, imamo

$r_{l-i} \geq \phi^i$. Tvrdnja slijedi postavljanjem $i = l - 1$ i korištenjem logaritma.
Sada dokazujemo gornju tvrdnju. Za $i = 0$ i $i = 1$, imamo

$$r_l \geq 1 = \phi^0 \qquad r_{l-1} \geq r_l + 1 \geq 2 \geq \phi^1.$$

Za $i = 2, \ldots, l - 1$, koristeći indukciju i činjenicu da je $\phi^2 = \phi + 1$, slijedi

$$r_{l-i} \geq r_{l-(i-1)} + r_{l-(i-2)} \geq \phi^{i-1} + \phi^{i-2} = \phi^{i-2}(1 + \phi) = \phi^i,$$

čime smo dokazali tvrdnju. □

Navedimo neke primjere koji će nam pokazati kako koristiti prethodni teorem.

Primjer 4. *Neka je $a = 95$ i $b = 35$. Sada se brojevi koji se pojavljuju u prethodnom teoremu lako mogu izračunati na sljedeći način:*

$$95 = 35 \cdot 2 + 25$$
$$35 = 25 \cdot 1 + 10$$
$$25 = 10 \cdot 2 + 5$$
$$10 = 5 \cdot 2 + 0.$$

i	0	1	2	3	4	5
r_i	95	35	25	10	5	0
q_i		2	1	2	2	

Iz tablice jasno vidimo da je $nzd(a, b) = 5$.

Postupak opisan u Teoremu 3. lako se može pretvoriti u jednostavni algoritam.
Euklidov algoritam. Neka su na ulazu cijeli brojevi a,b, takvi da $a \geq b \geq 0$, izračunajmo $d = nzd(a, b)$ prema sljedećem algoritmu ([3], stranica 75):

$r \leftarrow a, r' \leftarrow b$
while $r \neq 0$ do
    $r' \leftarrow r \mod r'$
    $(r, r') \leftarrow (r', r'')$
$d \leftarrow r$
output d

Sada ćemo promatrati vrijeme izvršenja Euklidovog algoritma. Intuitivno, to bi se moglo pokazati na sljedeći način. Neka su a i b k-bitni brojevi. Broj dijeljenja koje izvodi navedeni algoritam je broj l iz Teorema 3., koji ima vrijeme izvršenja $O(k)$. Nadalje, svako dijeljenje uključuje brojeve od k bitova ili manje pa je to vrijeme izvršenja $O(k^2)$, što dovodi do ograničenja vremena izvođenja od $O(k^3)$. Međutim, kao što pokazuje sljedeći teorem, ovo kubično ograničenje vremena izvođenja je daleko od željenog rezultata. Intuitivno, to je zato

što vrijednost izvođenja dijeljenja ovisi o duljini kvocijenta:
što je veći kvocijent, "skuplje" je dijeljenje.
Sada ćemo definirati *duljinu* broja a. Za cijeli broj a, definiramo njegovu bitnu duljinu, ili jednostavno, njegovu duljinu, koju označavamo s $len(a)$, tako da bude broj bitova binarnog prikaza $|a|$, točnije,

$$len(a) = \begin{cases} \lfloor \log_2 |a| \rfloor + 1 \,, & \text{ako je } a \neq 0, \\ 1 \qquad\qquad\quad\;\; , & \text{ako je } a = 0. \end{cases}$$

Ako je $len(a) = l$, tada kažemo da je a l-bitni broj. Primijetimo ako je a pozitivan, l-bitni broj, tada vrijedi $\log_2 a < l \leq \log_2 a + 1$, odnosno, $2^{l-1} \leq a < 2^l$.
Pogledajmo sada kakvo vrijeme izvršenja ima Euklidov algoritam.

Teorem 4 (vidjeti [3, Teorem 4.2.]). *Euklidov algoritam ima vrijeme izvršenja* $O(len(a)len(b))$.

Dokaz. Pretpostavimo da je $b > 0$. Uz oznake kao u Teoremu 3., vrijeme izvršenja je $O(T)$, pri čemu je

$$\begin{aligned} T &= \sum_{i=1}^{l} len(r_i)len(q_i) \leq len(b) \sum_{i=1}^{l} len(q_i) \\ &\leq len(b) \sum_{i=1}^{l} (len(r_{i-1}) - len(r_i) + 1) \\ &= len(b)(len(r_0) - len(r_l) + l) \\ &\leq len(b)(len(a) + len(b)/\log\phi + 1) \\ &= O(len(a)len(b)). \end{aligned}$$

Time je dokazana tvrdnja teorema. □

1.3 Prošireni Euklidov algoritam

Prema Teoremu 2. najveći zajednički djelitelj brojeva a i b može se prikazati kao cjelobrojna linearna kombinacija od a i b, tj. postoje $s, t \in \mathbb{Z}$ takvi da je

$$nzd(a, b) = as + bt.$$

Uz računanje $nzd(a, b)$, prošireni Euklidov algoritam računa i cjelobrojne koeficijente s, t. Sljedeći teorem definira vrijednosti izračunate ovim algoritmom te iznosi niz važnih činjenica o njima, a oni će imati ključnu ulogu, kako u analizi vremena izvođenja algoritma, tako i u primjenama algoritma o kojima ćemo raspravljati kasnije.

Teorem 5 (vidjeti [3, Teorem 4.3.]). *Neka su* $a, b, r_0, \ldots r_{l+1}$ *i* $q_1, \ldots, q_l$ *definirani kao u Teoremu 3. Definirajmo cijele brojeve* $s_0, \ldots, s_{l+1}$ *i* $t_0, \ldots, t_{l+1}$ *na sljedeći način:*

$$\begin{aligned} s_0 &:= 1, & t_0 &:= 0, \\ s_1 &:= 0, & t_1 &:= 1, \\ s_{i+1} &:= s_{i-1} - s_i q_i, & t_{i+1} &:= t_{i-1} - t_i q_i \qquad (i = 1, \ldots, l). \end{aligned}$$

Tada vrijedi

(*i*) *za* $i = 0, \ldots, l+1$, *vrijedi* $as_i + bt_i = r_i$; *posebno* $as_l + bt_l = nzd(a,b)$;

(*ii*) *za* $i = 0, \ldots, l$, *vrijedi* $s_i t_{i+1} - t_i s_{i+1} = (-1)^i$;

(*iii*) *za* $i = 0, \ldots, l+1$, *vrijedi* $nzd(a,b) = 1$;

(*iv*) *za* $i = 0, \ldots, l$, *vrijedi* $t_i t_{i+1} \leq 0$ *i* $|t_i| \leq |t_{i+1}|$; *a za* $i = 0, \ldots, l$, *vrijedi* $s_i s_{i+1} \leq 0$ *i* $|s_i| \leq |s_{i+1}|$;

(*v*) *za* $i = 0, \ldots, l+1$, *vrijedi* $r_{i-1}|t_i| \leq a$ *i* $r_{i-1}|s_i| \leq b$;

(*vi*) *ako je* $a > 0$, *tada za* $i = 0, \ldots, l+1$, *vrijedi* $|t_i| \leq a$ *i* $|s_i| \leq b$; *ako je* $a > 1$ *i* $b > 0$, *tada* $|t_l| \leq a/2$ *i* $|s_l| \leq b/2$.

Dokaz. (i) se lako dokaže matematičkom indukcijom po i. Za $i = 0, 1$, tvrdnja je očigledna. Za $i = 2, \ldots, l+1$ imamo

$$\begin{aligned} as_i + bt_i &= a(s_{i-2} - s_{i-1}q_{i-1}) + b(t_{i-2} - t_{i-1}q_{i-1}) \\ &= (as_{i-2} + bt_{i-2}) - (as_{i-1} + bt_{i-1})q_{i-1} \\ &= r_{i-2} - r_{i-1}q_{i-1} \\ &= r_i. \end{aligned}$$

(ii) se također lako može dokazati matematičkom indukcijom po i. Za $i = 0$, tvrdnja je očigledna. Za $i = 1, \ldots, l$, vrijedi

$$\begin{aligned} s_i t_{i+1} - t_i s_{i+1} &= s_i(t_{i-1} - t_i q_i) - t_i(s_{i-1} - s_i q_i) \\ &= -(s_{i-1}t_i - t_{i-1}s_i) \\ &= -(-1)^{i-1} \\ &= (-1)^i. \end{aligned}$$

(iii) slijedi direktno kao posljedica (ii).
Tvrdnje iz (iv) lako možemo dokazati matematičkom indukcijom po i. Tvrdnja koja uključuje t_i očigledno vrijedi za $i = 0$, a za $i = 1, \ldots, l$ vrijedi

$$t_{i+1} = t_{i-1} - t_i q_i,$$

a budući da po hipotezi indukcije vrijedi da t_{i-1} i t_i imaju suprotne predznake i $|t_i| \geq |t_{i-1}|$, slijedi $|t_{i+1}| = |t_{i-1}| + |t_i|q_i \geq |t_i|$, i predznaci od t_{i+1} i t_i su suprotni. Dokaz tvrdnje koja uključuje s_i se dokazaju analogno, osim što indukcija kreće od $i = 1$.
Kod dokazivanja tvrdnje (v), promatraju se dvije jednadžbe:

$$\begin{aligned} as_{i-1} + bt_{i-1} &= r_{i-1}, \\ as_i + bt_i &= r_i. \end{aligned}$$

Oduzimanjem t_{i-1} iz druge jednadžbe, a t_i iz prve te primjenom (ii), dobivamo $\pm a = t_i r_{i-1} - t_{i-1} r_i$; stoga, koristeći činjenicu da su t_i i t_{i-1} suprotnog predznaka, dobivamo

$$a = |t_i r_{i-1} - t_{i-1} r_i| = |t_i| r_{i-1} + |t_{i-1}| r_i \geq |t_i| r_{i-1}.$$

Nejednakost koja uključuje s_i dokazuje se na sličan način.
Tvrdnja (vi) slijedi iz (v) i sljedećih činjenica: ako je $a > 0$, tada je $r_{i-1} > 0$ za $i = 1, \ldots, l+1$; ako je $a > 1$ i $b > 0$, tada je $l > 0$ i $r_{l-1} \geq 2$. □

Primjer 5. *Koristit ćemo brojeve kao u Primjeru 4., tj. neka je* $a = 95$ *i* $b = 35$. *Sada se* t_i *i* s_i *lako izračunaju pomoću* q_i*:*

i	0	1	2	3	4	5
r_i	95	35	25	10	5	0
q_i		2	1	2	2	
s_i	1	0	1	6	3	7
t_i	0	1	-2	-16	-8	-19

Iz tablice se jasno vidi da je $nzd(a, b) = 5 = 3a - 8b$.

Postupak koji je opisan u Teoremu 5. može se pretvoriti u jednostavan algoritam.
Prošireni Euklidov algoritam. Na ulazu cijeli brojevi a, b, za koje vrijedi $a \geq b \geq 0$, izračunaj cijele brojeve d, s i t tako da $d = nzd(a, b)$ i $as + bt = d$, prema sljedećem ([3], stranica 78):

$r \leftarrow a, r' \leftarrow b$
$s \leftarrow 1, s' \leftarrow 0$
$t \leftarrow 0, t' \leftarrow 1$
while $r \neq 0$ do
    $q \leftarrow \lfloor r/r' \rfloor, r'' \leftarrow r \mod r'$
    $(r, s, t, r_0, s_0, t_0) \leftarrow (r', s', t', r'', s - s'q, t - t'q)$
$d \leftarrow r$
output d, s, t

Teorem 6 (vidjeti [3, Teorem 4.4.])**.** *Prošireni Euklidov algoritam ima vrijeme izvršenja* $O(len(a)len(b))$.

Dokaz. Pretpostavimo neka je $b > 0$. Dovoljno je analizirati vrijednosti lista $\{s_i\}$ i $\{t_i\}$. Promotrimo prvo sve vrijednosti od t_i čije je vrijeme izvršenja $O(T)$, pri čemu je $T = \sum_{i=1}^{l} len(t_i)len(q_i)$. Neka je $t_1 = 1$, prema tvrdnji (vi) Teorema 5., slijedi $|t_i| \leq a$ za

$i = 2, \ldots, l$. Analizirajući kao u dokazu Teorema 4., slijedi

$$\begin{aligned} T &\leq len(q_1) + len(a) \sum_{i=2}^{l} len(q_i) \\ &\leq len(a) + len(a)(len(r_1) - len(r_l) + l - 1) = O(len(a)len(b)). \end{aligned}$$

Analogno, za sve vrijednosti od s_i može se pokazati da je njihovo vrijeme izvršenja $O(len(a)len(b))$, odnosno $O(len(b)^2)$. □

Vrlo je važna i efikasnost Euklidova algoritma. Najprije ćemo promatrati broj koraka, odnosno broj dijeljenja, u algoritmu. Pokazat ćemo da je broj koraka proporcionalan broju znamenaka od b. Algoritmi u kojima je broj operacija proporcionalan nekoj potenciji broja znamenaka ulaznog podatka nazivaju se *polinomijalni* ili *efikasni algoritmi.*

Propozicija 1 (vidjeti [1, Propozicija 2.8.]). *Za broj koraka i u Euklidovom algoritmu vrijedi*

$$i < 2 \log_2 b.$$

Dokaz. Pogledajmo j-ti korak. Imamo $r_j \leq \frac{r_{j-1}}{2}$ ili $\frac{r_{j-1}}{2} < r_i < rj - 1$. Za drugi slučaj vrijedi $q_{j+1} = 1$ i $r_{j+1} = r_{j-1} - r_j < \frac{r_{j-1}}{2}$. Dakle, u svakom slučaju vrijedi $r_{j+1} < \frac{r_{j-1}}{2}$, odatle slijedi

$$1 \leq r_i < \frac{r_{i-2}}{2} < \frac{r_{i-4}}{2} < \cdots < \frac{r_0}{2^{i/2}}$$

ako je i paran, a

$$2 \leq r_{i-1} < \frac{r_{i-3}}{2} < \frac{r_{i-5}}{2} < \cdots < \frac{r_0}{2^{(i-1)/2}}$$

ako je i neparan. Dakle, u svakom slučaju vrijedi $b = r_o > 2^{i/2}$ pa primjenom logaritma dobivamo $i < 2 \log_2 b$. □

Uz navedeni Teorem 5., možemo koristiti i matrični zapis. Za $i = 1, \ldots, l$ imamo

$$\begin{pmatrix} r_i \\ r_{i+1} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & -q_i \end{pmatrix} \begin{pmatrix} r_{i-1} \\ r_i \end{pmatrix}.$$

Rekurzivno proširujući desnu stranu ove jednakosti, dobivamo

$$\begin{pmatrix} r_i \\ r_{i+1} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & -q_i \end{pmatrix} \cdots \begin{pmatrix} 0 & 1 \\ 1 & -q_i \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix},$$

pri čemu vrijedi da je

$$M_i = \begin{pmatrix} 0 & 1 \\ 1 & -q_i \end{pmatrix} \cdots \begin{pmatrix} 0 & 1 \\ 1 & -q_i \end{pmatrix}.$$

Tako su definirane 2×2 matrice M_i za $i = 1, \ldots, l$. Ako definiramo M_0 kao 2×2 jediničnu matricu, lako je uočiti da za $i = 0, \ldots, l$ vrijedi

$$M_i = \begin{pmatrix} s_i & t_i \\ s_{i+1} & t_{i+1} \end{pmatrix}.$$

Iz svega prethodno navedenog, zaključujemo da tvrdnja (i) Teorema 5. direktno slijedi, a tvrdnja (ii) slijedi iz činjenice da je matrica M_i produkt i matrica čije su determinante jednake -1, stoga je očigledno da je determinanta matrice M_i jednaka $s_i t_{i+1} - s_{i-1} t_i$.

Primjer 6. *Neka su* $a = 162$, $b = 114$ *te odredimo* $nzd(a, b)$, s *i* t *pomoću matrice* M_i.

Rješenje:
Pomoću Euklidovog algoritma dobivamo

$$q_1 = 1, q_2 = 2, q_3 = 2, q_4 = 1, q_5 = 2, d = 2.$$

Sada je

$$M_4 = \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & -2 \end{bmatrix} = \begin{bmatrix} -7 & 19 \\ 10 & -27 \end{bmatrix}$$

pa slijedi da je $s = 19$, $t = -27$, a $nzd(162, 114) = 6$.

2 Primjene Kineskog teorema o ostacima i Fermatovog teorema

Osim Euklidovog algoritma, važnu ulogu u teoriji brojeva imaju i Kineski teorem o ostacima te Fermatov teorem. Sada ćemo opširnije objasniti njihovo značenje i kakva je njihova primjena prilikom stvaranja novih algoritama.

2.1 Računanje modularnih inverza i Kineski teorem o ostacima

U ovom poglavlju uvest ćemo osnovne pojmove vezane uz linearne kongruencije i njena rješenja. Osim toga, nešto više ćemo reći o Kineskom teoreomu o ostatcima te njegovoj primjeni i vremenu izvođenja.

Neka su $a, b \in \mathbb{Z}, n \in \mathbb{N}$. Rješavanje linearne kongruencije

$$ax \equiv b \pmod{n}$$

znači pronalaženje svih cijelih brojeva x koji zadovoljavaju navedenu kongruenciju. Pogledajmo sada specijalnu linearnu kongruenciju

$$ax \equiv 1 \pmod{n}.$$

Ta kongruencija ima rješenje ako i samo ako je $nzd(a, n) = 1$ te u tom slučaju ima jedinstveno rješenje u $\mathbb{Z}_n$. To rješenje nazivamo **modularni inverz** od a. Dakle, to rješenje je jedinstveni **multiplikativni inverz** od $a_0 \in \mathbb{Z}_n$, gdje je $a_o \equiv a \pmod{n}$.

Jedna od primjena proširenog Euklidovog algoritma je upravo računanje multiplikativnih inverza u $\mathbb{Z}_n$. Pretpostavimo da je $n > 1$. Dan nam je broj $b \in \{0, \ldots, n-1\}$, u vremenu $O(len(n)^2)$, pri čemu možemo odrediti je li b relativno prost s n. Ako vrijedi da su b i n relativno prosti, možemo izračunati $b^{-1} \mod n$ prema sljedećim koracima. Pokretanjem proširenog Euklidovog algoritma, na ulazu su n, b, dobivamo cijele brojeve d, s i t, takvi da $d = nzd(n, b)$ te $ns + bt = d$. Ako je $d \neq 1$, tada b nema multiplikativnih inverza modulo n. Inače, ako je $d = 1$, tada je t multiplikativni inverz od b modulo n; međutim, možda se neće nalaziti u traženom skupu $\{0, \ldots, n-1\}$. Prema tvrdnji (vi) iz Teorema 5., vrijedi $|t| \leq n/2 < n$; stoga je $t \in \{0, \ldots, n-1\}$ ili $t < 0$ i $t + n \in \{0, \ldots, n-1\}$. Dakle b^{-1} modulo n jednak je ili t ili $t + n$.
Kineski teorem o ostacima daje informacije o rješenju sustava linearnih kongruencija. Ime je dobio zbog povezivanja s kineskim matematičarem iz prvog stoljeća Sun Tzu, a dokazan je tek u 13. stoljeću. Prva primjena teorema je bila u kineskoj vojsci za prebrojavanje vojnika.

Teorem 7 (Kineski teorem o ostacima,vidjeti [1, Teorem 3.7.]). *Neka su $n_1, \ldots, n_k$ u parovima relativno prosti brojevi te neka su $a_1, a_2, \ldots, a_k$ cijeli brojevi. Tada sustav kongruencija*

$$x \equiv a_1 \pmod{n_1},\ x \equiv a_2 \pmod{n_2}, \ldots, x \equiv a_k \pmod{n_k}$$

ima rješenja. Ako je x_0 jedno rješenje, onda su sva rješenja dana s $x \equiv x_0 \pmod{n_1 n_2 \cdots n_k}$.

Dokaz. Pretpostavimo da je $n = n_1 n_2 \cdot \ldots \cdot n_k$ te $m_j = \frac{n}{n_j}$ za $j = 1, \ldots, k$. Zbog uvjeta da su $n_1, \ldots, n_k$ u parovima relativno prosti, slijedi $nzd(n_j, m_j) = 1$ pa postoji cijeli broj x_j takav da $m_j x_j \equiv a_j \pmod{n_j}$. Pogledajmo broj

$$x_0 = m_1 x_1 + \cdots + m_k x_k.$$

Za x_0 vrijedi

$$x_0 \equiv 0 + \cdots + 0 + m_j x_j + 0 + \cdots + 0 \equiv a_j \pmod{n_j}.$$

Dakle, možemo zaključiti da je x_0 rješenje sustava kongruencija iz tvrdnje teorema.
Ako su x, x_0 dva rješenja sustava, onda vrijedi da je $x \equiv x_0 \pmod{n_j}$ za $j = 1, \ldots, k$, a budući da su n_j u parovima relativno prosti, slijedi da je $x \equiv x_0 \pmod{m}$. □

Primjer 7. *Riješimo sljedeći sustav kongruencija*

$$\begin{aligned} x &\equiv 2 \pmod 3, \\ x &\equiv 3 \pmod 5, \\ x &\equiv 2 \pmod 7. \end{aligned}$$

Rješenje:
Znamo da $nzd(3,5) = nzd(5,7) = nzd(3,7) = 1$, tj. u parovima su relativno prosti, pa smijemo primijeniti Kineski teorem o ostacima. Vrijedi da je
$n = 3 \cdot 5 \cdot 7$,
$m_1 = 35, \quad m_2 = 21, \quad m_3 = 15.$
Sada imamo linearne kongruencije

$$\begin{aligned} 35x_1 \equiv 2 \pmod 3 &\iff 2x_1 \equiv 2 \pmod 3 \implies x_1 \equiv 1 \pmod 3 \\ 21x_2 \equiv 3 \pmod 5 &\iff x_2 \equiv 3 \pmod 5 \\ 15x_3 \equiv 2 \pmod 7 &\iff x_3 \equiv 2 \pmod 7 \end{aligned}$$

Uzmimo sada $x_1 = 2, x_2 = 3, x_3 = 2$ te slijedi

$$x \equiv 2 \cdot 35 + 1 \cdot 21 + 3 \cdot 15 \pmod{105},$$

$$x \equiv 128 \pmod{105},$$

$$x \equiv 23 \pmod{105}.$$

Također, možemo primijetiti kako se Kineski teorem o ostacima može napisati kao efikasan računalni algoritam.

Teorem 8 (vidjeti [3, Teorem 4.5.]). *Neka su nam dani cijeli brojevi $n_1, \ldots, n_k$ i $a_1, \ldots, a_k$, pri čemu su $\{n_i\}_{i=1}^k$ po parovima relativno prosti, te neka su $n_i > 1$ i $0 \leq a_i < n_i$ za $i = 1, \ldots, k$. Neka je $n := \prod_{i=1}^k n_i$. Tada je vrijeme izvođenja $O(len(n)^2)$, može se izračunati jedinstveni cijeli broj a koji zadovoljava $0 \leq a < n$ i $a \equiv a_i \pmod{n_i}$ za $i = 1, \ldots, k$.*

Dokaz prethodnog teorema je izravna implementacija sljedećeg algoritma:

$n \leftarrow \prod_{i=1}^{k} n_i$
for $i \leftarrow 1$ to k do
$n_i^* \leftarrow n/n_i; b_i \leftarrow n_i^* \mod n_i, t_i \leftarrow b_i^{-1} \mod n_i, e_i \leftarrow n_i^* t_i$
$a \leftarrow (\sum_{i=1}^{k} a_i e_i) \mod n$

2.2 Ubrzanje algoritma pomoću modularne aritmetike

Važna praktična primjena gornje "računalne" verzije Kineskog teorema o ostacima je opća algoritamska tehnika koja može značajno ubrzati određene vrste računanja koja uključuju višeznamenkaste cijele brojeve. Umjesto da pokušamo opisati tehniku u nekom općenitom obliku, jednostavno ćemo ilustrirati objašnjenje tehnike pomoću specifičnog primjera, a to je množenje matrica čiji su elementi cijeli brojevi.

Pretpostavimo da imamo dvije $m \times m$ matrice A i B, te da su njihovi elementi veliki cijeli brojevi. Sada želimo izračunati umnožak matrica $C = AB$. Pretpostavimo da je za $r, s = 1, \ldots, m$, element matrice A u retku r i stupcu s jednak a_{rs}, te za $s, t = 1, \ldots, m$, element matrice B u retku s i stupcu t jednak b_{st}. Neka je sada za $r, t = 1, \ldots, m$ element matrice C u retku r i stupcu t jednak c_{rt}, koji je dobiven standardnim pravilom množenja matrica:

$$c_{rt} = \sum_{i=1}^{m} a_{rs} b_{st}.$$

Nadalje, pretpostavimo da je M najveća apsolutna vrijednost elemenata u A i B, tako da su elementi u C ograničeni po apsolutnoj vrijednosti zbog $M' = M^2 m$. Neka je $l := len(M)$. Radi pojednostavljenja, pretpostavimo da vrijedi $m \leq M$ (ovo je razumna pretpostavka, budući da želimo uzeti u obzir velike vrijednosti matrice M, recimo veće od 2^{100}, svakako ne možemo očekivati da ćemo lako računati s $2^{100} \times 2^{100}$ matricama).
Koristeći prethodno navedenu formulu za računanje elemenata matrice C, možemo izračunati elemenete matrice C koristeći m^3 množenja brojeva duljine najviše l te m^3 zbrajanja brojeva duljine najviše $len(M')$, pri čemu je $len(M') \leq 2l + len(m) = O(l)$. Iz ovoga slijedi vrijeme izvršenja

$$O(m^3 l^2).$$

Koristeći Kineski teorem o ostacima, možemo postići puno bolji rezultat od prethodno navedenog.

Za svaki $n > 1$, te za sve $r, t = 1, \ldots, m$, imamo

$$c_{rt} \equiv \sum_{s=1}^{m} a_{rs} b_{st} \pmod{n}.$$

Nadalje, ako izračunamo c'_{rt} kao

$$c'_{rt} \equiv \sum_{s=1}^{m} a_{rs} b_{st} \pmod{n}$$

te ako uz to vrijedi

$$-n/2 \leq c'_{rt} < n/2 \qquad i \qquad n > 2M',$$

tada sigurno vrijedi

$$c_{rt} = c'_{rt}.$$

Kako bismo se uvjerili da prethodna jednakost zaista vrijedi, moramo primijetiti kako iz prve dvije kongruencije slijedi $c_{rt} \equiv c'_{rt} \pmod{n}$, što znači da n dijeli $c_{rt} - c'_{rt}$. Zatim uz danu nejednakost $|c_{rt}| \leq M'$, zaključujemo

$$|c_{rt} - c'_{rt}| \leq |c_{rt}| + |c'_{rt}| \leq M' + n/2 < n/2 + n/2 = n.$$

Dakle, možemo zaključiti kako je razlika $c_{rt} - c'_{rt}$ višekratnik od n, dok je istovremeno ta razlika manja od n po apsolutnoj vrijednosti. Dakle, ova razlika mora biti jednaka nuli, čime je dokazano da vrijedi $c_{rt} = c'_{rt}$.
Stoga, da bi se izračunali elemente matrice C, dovoljno je izračunati elemente od C modulo n, pri čemu moramo biti sigurni da su izračunati ostaci u intervalu $[-n/2, n/2\rangle$.

Kako bismo izračunali C modulo n, biramo niz manjih brojeva $n_1, \ldots, n_k$, takvih da su $\{n_i\}_{i=1}^k$ po parovima relativno prosti, te neka je $n = \prod_{i=1}^k n_i$ samo malo veći od $2M'$. U praksi bi se odabrali n_i tako da su mali prosti brojevi, a tablica takvih prostih brojeva mogla bi se lako izračunati unaprijed, tako da se mogu obraditi svi problemi do zadane veličine. Primjerice, umnožak svih prostih brojeva, od najviše 16 bitova, je broj koji ima više od 90000 bitova. Prema tome, jednostavnim prethodnim računanjem i pohranjivanjem takvih tablica malih prostih brojeva mogu se riješiti ulazne matrice s prilično velikim unosima (do oko 45000 bitova).
Pretpostavimo da smo unaprijed izračunali odgovarajuće proste brojeve $n_1, \ldots, n_k$. Nadalje, pretpostavit ćemo da zbrajanje i množenje modulo n_i može biti dovršeno u *konstantnom* vremenu. To je razumno s praktičnog (i teorijskog) gledišta, budući da se takvi prosti brojevi lako "uklapaju" u strojne izraze i možemo izvesti modularno zbrajanje i množenje pomoću konstantnog broja operacija stroja. Konačno, pretpostavljamo da ne koristimo više n_i nego što je potrebno, dakle vrijedi $len(n) = O(l)$ i $k = O(l)$.

Da bismo izračunali C, izvršavamo sljedeće korake:

1) Za svaki $i = 1, \ldots, k$, učini sljedeće:

   (a) izračunaj $a_{rs}^{(i)} \leftarrow a_{rs} \mod n_i$, za $r, s = 1, \ldots, m$,

   (b) izračunaj $b_{st}^{(i)} \leftarrow b_{st} \mod n_i$, za $r, s = 1, \ldots, m$,

   (c) Za $r, t = 1, \ldots, m$, izračunaj

$$c_{rt}^{(i)} \leftarrow \sum_{i=1}^{m} a_{rs}^{(i)} b_{st}^{(i)} \mod n_i.$$

2) Za svaki $r, t = 1, \ldots, m$, primijeni Kineski teorem o ostacima na $c_{rt}^1, c_{rt}^2, \ldots, c_{rt}^k$ te dobivamo c_{rt}, koji treba izračunati kao "uravnoteženi" ostatak modulo n, tj. $-n/2 \leq c_{rt} < n/2$.

3) Ispiši matricu C, čiji je element u retku s i stupcu t jednak c_{rt}.

Primijetimo kako u koraku 2, ako je algoritam Kineskog teorema o ostacima implementiran tako da računa $a \in \mathbb{Z}$ takav da je $0 \leq a < n$, lako možemo izračunati "uravnoteženi" ostatak samo oduzimanjem n od a, ako je $a > n/2$.
Ispravnost gornjeg algoritma je već utvrđena. Analizirajmo sada njegovo vrijeme izvršenja. Može se lako vidjeti da je vrijeme izvršenja koraka $1a$ i $1b$ jednako $O(m^2l^2)$. Pod našom pretpostavkom o vremenu izvršenja za modulo prostih brojeva, vrijeme izvršenja koraka $1c$ je $O(m^3k)$, a budući da je $k = O(l)$, vrijedi da je vrijeme izvršenja ovog koraka $O(m^3l)$. Naposljetku, prema Teoremu 8., vrijeme izvršenja koraka 2 je $O(m^2l^2)$. Dakle, konačno vrijeme izvršenja ovog algoritma je

$$O(m^2l^2 + m^3l).$$

Ovo je značajan napredak u usporedbi s $O(m^3l^2)$. Primjerice, ako je $l \approx m$, tada je vrijeme izvršenja početnog algoritma $O(m^5)$, dok je vrijeme izvršenje modularnog algoritma $O(m^4)$.

2.3 Fermatov teorem i njegova efikasnost

Za neparne proste brojeve oblika $4k + 1$ vrijedi Fermatov teorem o zbroju dva kvadrata. Sada ćemo dokazati Thueovu lemu jer će nam ona biti potrebna u dokazu Fermatovog teorema i u daljnjem proučavanju efikasnosti.

Teorem 9 (Thueova lema, vidjeti [3, Teorem 2.33.]). *Neka su $n, b, r^*, t^* \in \mathbb{Z}$ takvi da $0 < r \leq n < r^*t^*$. Tada postoje $r, t \in \mathbb{Z}$ takvi da vrijedi*

$$r \equiv bt \pmod{n}, \quad |r| < r^*r, \quad 0 < |t| < t^*.$$

Dokaz. Za $i = 1, \ldots, r^* - 1$ i $j = 1, \ldots, t^* - 1$, definiramo $v_{ij} = i - bj$. Budući da smo definirali r^*t^* brojeva pri čemu je $r^*t^* > n$, neka dva broja moraju se nalaziti u istoj klasi ostataka modulo n. Odnosno, za neke $(i_1, j_1) \neq (i_2, j_2)$, imamo $v_{i_1j_1} \equiv v_{i_2j_2} \pmod{n}$. Neka je $r = i_1 - i_2$ te $t = j_1 - j_2$, odakle slijedi da je $r \equiv bt \pmod{n}, |r| < r^*, |t| < t^*$, te također $r \neq 0$ ili $t \neq 0$. Preostalo je još pokazati da je $t \neq 0$. Pretpostavimo suprotno, tj. neka je $t = 0$. To bi značilo da je $r \equiv 0 \pmod{n}$ i $r \neq 0$, što bi značilo da je r višekratnik od n različit od 0, međutim, to je nemoguće jer je $|r| < r^* \leq n$. □

Budući da smo dokazali potrebnu lemu, prelazimo na dokaz Fermatovog teorema o zbroju dva kvadrata.

Teorem 10 (vidjeti [3, Teorem 2.34.]). *Neparan prost broj p se može prikazati kao suma dva kvadrata ako i samo ako je p oblika $4k + 1$.*

Dokaz. Dokaz jednog smjera je jednostavan. Pretpostavimo da je $p \equiv 3 \pmod{4}$. Uočavamo kako je kvadrat svakoga broja kongruentan 0 ili 1 modulo 4, dakle, zbroj dva kvadrata je kongruentan 0, 1 ili 2 modulo 4 pa ne mogu biti kongruentni p modulo 4.

Sada kako bismo dokazali drugi smjer, pretpostavimo da je $p \equiv 1 \pmod 4$. Znamo da je -1 kvadratni ostatak modulo p pa neka je b takav da $b^2 \equiv -1 \pmod p$. Sada primjenjujući Teorem 9. na $n = p$, pri čemu je b kako smo upravo definirali, $r^* = t^* = \lfloor \sqrt{p} \rfloor + 1$. Očito, $\lfloor \sqrt{p} \rfloor + 1 > \sqrt{p}$, stoga vrijedi $r^* t^* > p$. Također, budući da je p prost, $\sqrt{p}$ nije cijeli broj, pa je $\lfloor \sqrt{p} \rfloor < \sqrt{p} < p$, posebno, $r^* = \lfloor \sqrt{p} \rfloor + 1 \leq p$. Dakle, ispunjeni su uvjeti Thueove leme pa postoje cijeli brojevi r i t takvi da

$$r \equiv bt \pmod p, |r| \leq \lfloor \sqrt{p} \rfloor < \sqrt{p}, 0 < |t| \leq \lfloor \sqrt{p} \rfloor < \sqrt{p}.$$

Slijedi

$$r^2 \equiv b^2 t^2 \equiv -t^2 \pmod p.$$

Dakle, $r^2 + t^2$ je višekratnik od p te $0 < r^2 + t^2 < 2p$. Jedina mogućnost je da je $r^2 + t^2 = p$. □

Navest ćemo primjer kako bismo provjerili Fermatov teorem.

Primjer 8. *Neka nam je dan prost broj $p = 17$. Vidimo da je taj broj oblika $4k + 1$, tj. $p = 17 = 4 \cdot 4 + 1$. Dakle, $p | a^2 + b^2$ za $a = \left(\frac{17-1}{2}\right)! = 8!$, $b = 1$. Kako je $8! \equiv 13 \pmod{17}$, slijedi da*

$$p \mid 13^2 + 1^2$$

i $10p = 13^2 + 1^2$, tj. $k = 10, x = 13, y = 1$. Sada tražimo najmanje ostatke od x i y pri dijeljenju s k. To su $u = 3, v = 1$. Neka su

$$x_1 = \frac{xu + yv}{k} = 4, y_1 = \frac{xv - yu}{k} = 1$$

te dobivamo $17 = 4^2 + 1^2$.

U ovom poglavlju dokazujemo da je ovaj teorem računalno efikasan, odnosno razvijamo efikasan algoritam koji kao ulazne podatke uzima proste brojeve oblika $p \equiv 1 \pmod 4$, a ispisuje cijele brojeve s, t takve da vrijedi $p = r^2 + t^2$. Bitnu ulogu u dokazivanju Fermatovog teorema ima Thueova lema. Thuova lema potvrđuje postojanje brojeva iz Fermatovog teorema, a to smo dokazali koristeći Dirichletov princip, koji se nažalost ne može izravno napisati kao učinkovit algoritam za pronalaženje takvih brojeva.

Međutim, može se pokazati da ti brojevi nastaju kao posljedica proširenog Euklidovog algoritam. Da bismo to učinili preciznijim, uvedimo neke oznake. Za $a, b \in \mathbb{Z}$, $a \geq b \geq 0$, definirajmo

$$EEA(a, b) = \{(r_i, s_i, t_i)\}_{i=1}^{l+1},$$

pri čemu su r_i, s_i i t_i, $i = 1, \ldots, l + 1$, definirani kao u Teoremu 5. Sada konačno možemo dokazati efikasnu Thuevu lemu.

Teorem 11 (Efikasna Thueova lema, vidjeti [3, Teorem 4.7.])**.** *Neka su $n, b, r^*, t^* \in \mathbb{Z}$, takvi da $0 \leq b < n$ te $0 < r^* \leq n < r^* t^*$. Nadalje, neka je $EEA(n, b) = \{(r_i, s_i, t_i)\}_{i=1}^{l+1}$ i neka je j najmanji indeks (među $0, \ldots, l + 1$) tako da vrijedi $r_j < r^*$. Zatim, postavimo $r = r_j$ i $t = t_j$, tada vrijedi*

$$r \equiv bt \pmod n, \quad 0 \leq r < r^*, \quad 0 < |t| < t^*.$$

Dokaz. Budući da je $r_0 = n \geq r^* > 0 = r_{l+1}$, vrijednost indeksa j je dobro definirana, štoviše, $j \geq 1$ i $r_{j-1} \geq r^*$. Slijedi

$$\begin{aligned} |t_j| &\geq n/r_{j-1} \\ &\geq n/r^* \\ &< t^* \qquad (n < r^* t^*). \end{aligned}$$

Kako vrijedi $j \geq 1$, prema tvrdnji (iv) Teorema 5., dobivamo $|t_j| \geq |t_1| > 0$. S obzirom da vrijedi $r_j = ns_j + bt_j$, slijedi tvrdnja teorema $r_j \equiv bt_j \pmod{n}$. □

Ono što nam ovaj teorem govori jest da uz dane n, b, r^*, t^*, kako bismo pronašli željene vrijednosti od r i t, moramo pokrenuti Euklidov algoritam s ulazom n, b. Tada dobivamo niz ostataka $r_0 > r_1 > r_2 > \ldots$, gdje je $r_0 = n$ i $r_1 = b$. Ako je r_j prvi ostatak u nizu koji je manji od r^* te ako su s_j i t_j odgovarajući brojevi dobiveni proširenim Euklidovim algoritmom, tada su $r = r_j$ i $t = t_j$ uspješno obavili svoj zadatak.
Drugu bitnu ulogu u dokazivanju Fermatvog teorema je tvrdnja koja osigurava postojanje kvadratnog ostatka $p \equiv -1$, kada je p oblika $4k + 1$. Potreban nam je i efikasan oblik ove tvrdnje. Pretpostavimo da za prost broj p vrijedi $p \equiv 1 \pmod{4}$, te želimo izračunati $\beta \in \mathbb{Z}_p^*$ takav da $\beta^2 = -1$. Dovoljno je pronaći $\gamma \in \mathbb{Z}_p^* \setminus (\mathbb{Z}_p^*)^2$, budući da tada $\beta = \gamma^{(p-1)/4}$ zadovoljava $\beta^2 = -1$. Iako ne postoji poznat efikasan, deterministički algoritam za pronalaženje takvog γ, znamo da pola elemenata od $\mathbb{Z}_p^*$ su kvadrati, a ostalih pola nije, što nam sugerira sljedeću jednostavnu strategiju "pokušaja i pogrešaka" kako bismo izračunali β ([3], stranica 86):

repeat
    choose $\gamma \in \mathbb{Z}_p^*$
    compute $\beta \leftarrow \gamma^{(p-1)/4}$
until $\beta^2 = -1$
output β.

Kao algoritam, nije u potpunosti određen, moramo odrediti proceduru za odabir γ u svakoj iteraciji. Razumno rješenje bi bilo jednostavno odabrati nasumičan γ, pri čemu bi to bio primjer *vjerojatnosnog algoritma*. Pretpostavimo da to ima smisla s matematičkog i algoritamskog gledišta, tj. za svaku iteraciju vrijedi da imamo 50% šanse za odabir "dobrog" γ, odnosno, jednog koji se ne nalazi u $\mathbb{Z}_p^*$. Iz ovoga slijedi da bismo s velikom vjerojatnošću trebali pronaći "dobrog" γ u samo nekoliko iteracija (vjerojatnost da nakon t ponavljanja nismo pronašli odgovarajući γ jednaka je $1/2^t$), te da je *očekivani* broj iteracija 2. Dakle, može se zaključiti da je vrijeme izvođenja ovog algoritma $O(len(p)^3)$.

Konačno, iskoristimo sve navedene pretpostavke kako bismo dobili algoritam za pronalaženje r, t, takvih da $p = r^2 + t^2$.

1. Pronađi $\beta \in \mathbb{Z}_p^*$, takav da je $\beta^2 = -1$, koristeći gornju strategiju "pokušaja i pogrešaka".

2. Postavimo da je b ostatak pri dijeljenju od β s p.

3. Pokrenuti prošireni Euklidov algoritam na ulazu p, b, kako bismo dobili $EEA(p, b)$ i primijeniti Teorem 11. na $n = p, b$, $r^* = t^* = \lfloor \sqrt{p} \rfloor + 1$ da bismo dobili vrijednosti od r i t.

4. Ispiši r, t.

Kada navedeni algoritam završi, dobivamo $r^2 + t^2 = p$, kao i što je traženo. Kako smo prethodno tvrdili da vrijedi $r \equiv bt \pmod{p}$ i $b^2 \equiv -1 \pmod{p}$, slijedi $r^2 + t^2 \equiv 0 \pmod{p}$, a budući da je $0 < r^2 + t^2 < 2p$, mora vrijediti $p = r^2 + t^2$. Očekivano vrijeme izvršenja koraka 1 je $O(len(p)^3)$. Vrijeme izvršenja koraka 3 je $O(len(p)^2)$. Dakle, ukupno (očekivano) vrijeme izvršenja je $O(len(p)^3)$.

Primjer 9. *Neka nam je prost broj* $p = 1009$*, lako vidimo da vrijedi* $p \equiv 1 \pmod{4}$*. Izrazimo* p *kao zbroj dvaju kvadrata prema prethodnom algoritmu. Najprije, moramo pronaći* x *takav da je* $x^2 \cong -1 \pmod{p}$*. Pokušajmo s nekim nasumičnim brojem, recimo* 17*, te ga potencirajmo na* $(p-1)/4 = 252$*. Možemo izračunati* $17^{252} \equiv 469 \pmod{1009}$ *i* $469^2 \equiv -1 \pmod{1009}$*. Vidimo da smo imali sreće s prvim pokušajem. Sada pokrenimo prošireni Euklidov algoritam s ulazima* $p = 1009$ *i* $b = 469$ *te dobivamo sljedeće podatke:*

i	r_i	q_i	s_i	t_i
0	1009		1	0
1	469	2	0	1
2	71	6	1	-2
3	43	1	-6	13
4	28	1	7	-15
5	15	1	-13	28
6	13	1	20	-43
7	2	6	-33	71
8	1	2	218	-469
9	0		-469	1009

Prvi r_j *koji je manji od granice* $r^* = \lfloor \sqrt{1009} \rfloor + 1 = 32$*, dobije se u koraku* $j = 4$*, pa postavimo* $r = 28$ *i* $t = -15$*. Time potvrđujemo da vrijedi* $r^2 + t^2 = 28^2 + 15^2 = 1009 = p$*.*

3 Racionalna rekonstrukcija i primjene

U prethodnom poglavlju vidjeli smo kako primijeniti prošireni Euklidov algoritam da bismo dobili efikasnu verziju Thueove leme. Nadalje, Thueova lema tvrdi kako za zadane cijele brojeve n, b, postoje (r, t) koji zadovoljavaju $r \equiv bt \pmod{n}$ te se nalaze u opisanom pravokutniku, pod uvjetom da je površina pravokutnika dovoljno velika u odnosu na n. U ovom poglavlju prvo dokazujemo teorem o jedinstvenosti, pod pretpostavkom da je površina pravokutnika nije prevelika. Naravno, vrijedi ako je $r \equiv bt \pmod{n}$, tada za svaki cijeli broj $q \neq 0$, imamo $rq \equiv b \pmod{tq}$ pa možemo samo pretpostavljati da je omjer r/t jedinstven. Nakon dokazivanja ovog teorema o jedinstvenosti, pokazat ćemo kako ovaj teorem predstaviti na računalno učinkovit način, a zatim pokazati nekoliko vrlo zanimljivih primjena.

Sada ćemo dokazati osnovni teorem o jedinstvenosti.

Teorem 12 (vidjeti [3, Teorem 4.8.]). *Neka su $n, b, r^*, t^* \in \mathbb{Z}$ takvi da $r^* \geq 0, t^* > 0$ te $n > 2r^*t^*$. Nadalje, pretpostavimo da $r, t, r', t' \in \mathbb{Z}$ zadovoljavaju*

$$r \equiv bt \pmod{n}, |r| < r^*, 0 < |t| < t^*,$$

$$r' \equiv bt' \pmod{n}, |r'| < r^*, 0 < |t'| < t^*.$$

Tada vrijedi da $r/t = r'/t'$.

Dokaz. Promatrat ćemo dvije kongruencije

$$r \equiv bt \pmod{n},$$
$$r' \equiv bt' \pmod{n}.$$

Izlučivanjem t i t', dobivamo

$$rt' - r't \equiv 0 \pmod{n}.$$

Međutim, vrijedi i

$$|rt' - r't| \geq |r||t'| + |r'||t| \geq 2r^*t^* < n.$$

Dakle, $rt' - r't$ je višekratnik od n, ali manji od n po apsolutnoj vrijednosti pa je jedina mogućnost da je $rt' - r't = 0$, što znači da $r/t = r'/t'$. Time je dokazan teorem o jedinstvenosti. □

Sada pretpostavimo da su nam dani $n, b, r^*, t^* \in \mathbb{Z}$ kao u prethodnom teoremu. Štoviše, pretpostavimo da postoje $r, t \in \mathbb{Z}$ koji zadovoljavaju kongruencije iz dokaza teorema, ali vrijednosti od r, t su nam nepoznate. Primijetimo kako pod uvjetima Teorema 12., Thueova lema ne može sa sigurnošću tvrditi postojanje takvih r, t, ali neke naše primjene će dokazati njihovo postojanje. Željeli bismo pronaći $r', t' \in \mathbb{Z}$ tako da zadovoljavaju navedene kongruencije te ako ih pronađemo, tada prema teoremu znamo da $r/t = r'/t'$. To nazivamo **problem racionalne rekonstrukcije**. Ovaj problem možemo učinkovito riješiti pomoću proširenog Euklidovog algoritma. Naprotiv, kao i u slučaju efektivne Thueove leme, željene vrijednosti r' i t' pojavljuju se kao očekivana rješenja tog algoritma.

Sada ćemo dokazati problem racionalne rekonstrukcije.

Teorem 13 (vidjeti [3, Teorem 4.9.]). *Neka su $n, b, r^*, t^* \in \mathbb{Z}$ takvi da $0 \geq b < n$, $0 \geq r^* < n$ i $t^* > 0$. Nadalje, pretpostavimo da postoje $r, s, t \in \mathbb{Z}$ za koje vrijedi*

$$r = ns + bt, |r| \geq r^*, 0 < |t| \geq t^*.$$

Neka je $EEA(a, b) = \{(r_i, s_i, t_i)\}_{i=1}^{l+1}$, a neka je j najmanji indeks (među $0, 1, \ldots, l+1$) takav da $r_j < r^*$ i neka su

$$r' = r_j, \quad s' = s_j, \quad t' = t_j.$$

Tada vrijedi:

(*i*) $0 < |t'| \geq t^*$;

(*ii*) *ako je $n > 2r^*t^*$, tada za $q \in \mathbb{Z}, q \neq 0$ vrijedi*

$$r = r'q, \quad s = s'q, \quad t = t'q.$$

Dokaz. Budući da vrijedi $r_0 = n > r^* \geq 0 = r_{l+1}$, indeks j je dobro definiran, štoviše, $j \geq 1$ pa dobivamo sljedeće nejednakosti

$$0 \leq r_j < r_{j-1}, \quad 0 < |t_j|, \quad |r| \geq r^*, \quad 0 < |t| \geq t^*,$$

zajedno s jednakostima

$$r_{j-1} = ns_{j-1} + bt_{j-1}, \tag{1}$$
$$r_j = ns_j + bt_j, \tag{2}$$
$$r = ns + bt. \tag{3}$$

Sada pogledajmo tvrdnju (i). Naš cilj je dokazati da vrijedi

$$|t_j| \geq t^*.$$

Ovo je najteži dio dokaza. U tu svrhu definiramo

$$\epsilon := s_j t_{j-1} - s_{j-1} t_j, \mu := (t_{j-1}s - s_{j-1}t)/\epsilon, \nu := (s_j t - t_j s)/\epsilon.$$

Budući da je $\epsilon = \pm 1$, μ, ν su cijeli brojevi, štoviše, može se lako provjeriti da zadovoljavaju jednakosti

$$s_j\mu + s_{j-1}\nu = s, \tag{4}$$
$$t_j\mu + t_{j-1}\nu = t. \tag{5}$$

Sada ćemo ove jednakosti koristiti kako bismo proučili vrijednosti od ϵ, μ, ν. Promatramo tri slučaja:

(i) Pretpostavimo da je $\nu = 0$. Tada iz slučaja (2) slijedi da $t_j \mid t$, a budući da je $t \neq 0$, slijedi $|t_j| \geq |t| \geq t^*$.

(ii) Pretpostavimo da $\mu\nu < 0$. U ovome slučaju, budući da t_j i t_{j-1} imaju suprotne predznake, jednakost (5) implicira $|t| = |t_j\mu| + |t_{j-1}\nu| \geq |t_j|$, imamo $|t_j| \leq |t| \geq t^*$.

(iii) Preostao je samo slučaj kada je $\nu \neq 0$ i $\mu\nu \geq 0$. Tvrdimo da to nije moguće. Množenjem jednakosti (4) s n i (5) s b, dobivamo

$$r_j\mu + r_{j-1}\nu = r.$$

Ako je $\nu \neq 0$ te μ i ν imaju jednake predznake, prema tome slijedi da $|r| = |r_j\mu| + |r_{j-1}\nu| \geq r_{j-1}$. Dakle, vrijedi $r_{j-1} \leq |r| \leq r^*$, što je u kontradikciji s $r_{j-1} > r^*$.

Time smo dokazali nejednakost $|t_j| \geq t^*$. Sada ćemo dokazati drugu tvrdnju teorema, koja se kritički odnosi prema prethodnoj nejednakosti. Pretpostavimo da

$$n > 2r^*t^*.$$

Sada iz (2) i (3) dobivamo

$$\begin{aligned} r_j &\equiv bt_j \pmod{n}, \\ r &\equiv bt \pmod{n}. \end{aligned}$$

Koristeći prethodno navedene nejednakosti, vidimo da je tvrdnja Teorema 12. zadovoljena te možemo zaključiti

$$rt_j - r_jt = 0.$$

Oduzimanjem t_j puta (3) od t puta (2) i koristeći prethodnu jednakost, dobivamo $n(st_j - s_jt) = 0$, dakle,

$$st_j - s_jt = 0.$$

Odatle slijedi da $t_j \mid s_jt$, a budući da je $gcd(s_j, t_j) = 1$, mora vrijediti $t_j \mid t$. Stoga, $t = t_jq$ za neki q, pri čemu mora vrijediti $q \neq 0$ jer je $t \neq 0$. Zamjenom t_jq s t dobivamo $r = r_jq$ i $s = s_jq$, čime je dokazana tvrdnja (ii). □

3.1 Rekonstrukcija razlomka iz decimalnog zapisa

Poznato je da svaki realan broj ima decimalni zapis te da je taj decimalni zapis jedinstven, pod uvjetom da ne uključujemo brojeve s beskonačno decimala, oblika $1/10 = 0.1000\ldots = 0.099\cdots$. Sada pretpostavimo da Ana i Maja igraju igru. Ana je zamislila neki racionalni broj $z = s/t$, pri čemu su $s, t \in \mathbb{Z}$ takvi da $0 \leq s < t$, te otkriva Maji neke znamenke koje se nalaze na vodećim decimalnim mjestima. Cilj ove igre je da Maja pogodi o kojem broju se radi. Zanima nas može li Maja odrediti broj z.

Maja će uspjeti odrediti treženi broj ukoliko zna gornju granicu M od t te ukoliko joj Ana kaže dovoljan broj znamenki. Naravno, Maja još iz osnovne škole zna da je z konačno periodičan, a ako ima dovoljno znamenki koje se nalaze u zadanom periodu, može odrediti o kojem broju je riječ. Ova tehnika je prilično beskorisna u praksi jer duljina perioda može biti ogromna. Metoda o kojoj ćemo govoriti zahtijeva samo duljinu znamenki $O(len(M))$.

Pretpostavimo da je Ana zamislila broj z između 0 i 1 te Maji otkriva vodećih k znamenki. Tada, ako je

$$z = 0.z_1z_2z_3\dots$$

decimalni zapis od z, tada Ana govori Maji koje su znamenke $z_1, \dots, z_k$. Nadalje, ako je 10^k puno manje od M^2, tada broj z nije jednoznačno određen tim znamenkama, budući da postoji $\Omega(M^2)$ različitih racionalnih brojeva oblika s/t, $0 \leq s < t \leq M$. Međutim, ako je $10^k > 2M^2$, onda ne samo da je z jedinstveno određen, nego ga pomoću Teorema 13. Maja može i izračunati.
Sada ćemo opisati učinkovite algoritme i za Anu i za Maju, ali prije nego što to učinimo, napravit ćemo nekoliko općih činjenica o decimalnom proširenju od z. Neka je e proizvoljan nenegativan cijeli broj i pretpostavimo da je zapisan u obliku $z = 0.z_1z_2z_3\dots$. Promotrimo

$$10^e z = z_1 \dots z_e z_{e+1} z_{e+2} \dots$$

Slijedi da je

$$\lfloor 10^e z \rfloor = z_1 \dots z_e.0.$$

Budući da je $z = s/t$, neka je $r = 10^e s \mod t$, tada $10^e s = \lfloor 10^e z \rfloor t + r$ te dijeljenjem s t dobivamo $10^e z = \lfloor 10^e z \rfloor + r/t$, pri čemu je $r/t \in [0, 1)$. Nadalje,

$$\frac{10^e \mod t}{t} = 0.z_{e+1}z_{e+2}z_{e+3}\dots$$

Zatim, promotrimo Anin zadatak. Na temelju gornje rasprave, Ana može koristiti sljedeći jednostavan, iterativni algoritam ([3], stranica 92) za računanje $z_1, \dots, z_k$, za neki $k \geq 1$, nakon što je izabrala s i t:

$x_1 \leftarrow s$
for $i \leftarrow 1$ to k do
    $y_i \leftarrow 10x_i$
    $z_i \leftarrow \lfloor y_i/t \rfloor$
    $x_{i+1} \leftarrow y_i \mod t$
output $z_1, \dots, z_k$

Ispravnost algoritma slijedi iz činjenice da za svaki $i = 1, 2, \dots$, imamo $x_i = 10^{i-1}s \mod t$. Nadalje, za $e = i - 1$, dobivamo $x_i/t = 0.z_iz_{i+1}z_{i+2}\dots$. Stoga, neka su $e = 1$ i x_i/t u ulozi od z pa imamo $\lfloor 10x_i/t \rfloor = z_i$. Budući da je vrijeme izvršenja svake iteracije $O(len(M))$, konačno vrijeme izvršenja Aninog zadatka je $O(k \cdot len(M))$.
Naposljetku, pogledajmo Majino rješenje. Zadane su znamenke $z_1, \dots, z_k$ od broja $z = s/t$, uz gornju granicu M, $t \geq M$, sada Maja može izračunati z na sljedeći način:

1. Izračunaj $n \leftarrow 10^k$ i $b \leftarrow \sum_{i=1}^{k} z_i 10^{k-i}$.

2. Pokreni prošireni Euklidov algoritam na ulazu n, b, kako bismo dobili $EEa(n, b)$, pa primijeni Teorem 13. na n, b i $r^* = t^* = M$ i dobivamo vrijednosti r', s', t'.

3. Ispiši racionalni broj $-s'/t'$.

Analizirajmo sada algoritam, pretpostavljajući da je $10^k > 2M^2$.
Kako bismo dokazali ispravnost, moramo pokazati da $z = -s'/t'$. Neka je $e = k$ u $\lfloor 10^e z \rfloor = z_1 \cdots z_e.0$, tada imamo $b = \lfloor nz \rfloor = \lfloor ns/t \rfloor$. Nadalje, ako stavimo $r = ns \mod t$, tada dobivamo

$$r = ns - bt, \geq r < t \geq r^*, 0 < t \geq t^*, n > 2r^*t^*.$$

Slijedi da cijeli brojevi r', t' iz Teorema 13. zadovoljavaju jednakosti $s = s'q$ i $-t = t'q$, za neki q. Dakle, $s/t = s'/t'$. Dodatno, prošireni Euklidov algoritam jamči da je $nzd(s', t') = 1$, tako da ne samo da dobivamo traženi z, nego dobivamo z izražen u obliku neskrativog razlomka.
Konačno vrijeme izvršenja Majinog problema je $O(k^2)$.
Zaključujemo da Ana i Maja mogu uspješno odigrati ovu igru s k izabranih znamenki pri čemu je $k = O(len(M))$, te u ovom slučaju njihovi algoritmi imaju vrijeme izvršenja $O(len(M)^2)$.

Primjer 10 (vidjeti [3, Primjer 4.5.]). *Ana je izabrala cijele brojeve s, t takve da $0 \leq s < t \leq 1000$, te je rekla Maji broj $z = s/t$ ima sedam decimalnih mjesta, odakle će Maja moći izračunati koji je to z. Pretpostavimo da je $s = 511$ i $t = 710$. Tada je $s/t = 0.7197183098591549\ldots$. Maja je zapisala znamenke $7, 1, 9, 7, 1, 8, 3$, te izračunala $n = 10^7$ i $b = 7197183$. Pokrenula je prošireni Euklidov algoritam na ulazima n, b. Maja je primila podatke koji su navedeni u Tablici 1. Prvi r_j, koji je manji od gornje granice $r^* = 1000$, se pojavljujeu desetom koraku, tj. $j = 10$. Maja je iščitala iz tablice $s' = 511$ i $t' = -710$, odakle dobiva $z = -s'/t' = 511/710$.*

Još jedna zanimljiva činjenica koju možemo uočiti u Tablici 1., je ta da su razlomci $-s_i/t_i$ jako dobre procjene razlomka $b/n = 7197183/10000000$. Zaista, ako izračunamo pogreške od $b/n + s_i/t_i$, za $i = 1, \ldots, 5$, dobivamo (približno)

$$0.72, \qquad -0.28, \qquad 0.053, \qquad -0.03, \qquad 0.0054.$$

Dakle, možemo procijeniti "kompliciranu" razlomak $7197183/10000000$ pomoću "vrlo jednostavnog" razlomka $5/7$, uz apsolutnu pogrešku manju od 0.006.

i	r_i	q_i	s_i	t_i
0	10000000		1	0
1	7197183	1	0	1
2	2802817	2	1	-1
3	1591549	1	-2	3
4	1211268	1	3	-4
5	380281	3	-5	7
6	70425	5	18	-25
7	28156	2	-95	132
8	14113	1	208	-289
9	14043	1	-303	421
10	70	200	511	-710
11	43	1	-102503	142421
12	27	1	103014	-143131
13	16	1	-205517	285552
14	11	1	308531	-428683
15	5	2	-514048	714235
16	1	5	1336627	-1857153
17	0		-7197183	10000000

Tablica 1: Majini podaci iz proširenog Euklidovog algoritma.

3.2 Pogreške Kineskog teorema o ostacima

Jedna od interpretacija Kineskog teorema o ostacima je pronalaženje cijelog broja a, pri čemu je $0 \leq a < n$, pomoću niza $(a_1, \ldots, a_k$, za koji vrijedi $a_i = a \mod n$, za $i = 1, \ldots, k,$. Tada učinkovito možemo pronaći a. Napomenimo dodatno kako vrijedi i $n = n_1 \cdots n_k$ te $n_1, \ldots, n_k$ u parovima relativno prosti.
Pretpostavimo da Ana šifrira a kao $(a_1, \ldots, a_k)$ te šalje šifru Maji putem neke komunikacijske mreže. Međutim, budući da mreža nije savršena, tijekom prijenosa šifre, neke (ali nadamo se ne previše njih) od vrijednosti $a_1, \ldots, a_k$ mogu biti neispravne. Pitanje je, može li Maja i dalje efikasno pronaći traženi a iz njegove oštećene šifre.

Da bi problem bio precizniji, pretpostavimo da je tražena, točna šifra od a jednaka $(a_1, \ldots, a_k)$ te neka je neispravna šifra jednaka $(b_1, \ldots, b_k)$. Neka je $G \subseteq \{1, \ldots, k\}$ skup svih "loših" indekasa i takvih da $a_i \neq b_i$. Pretpostavit ćemo da $|B| \leq l$, pri čemu je l neki unaprijed određen parametar. Nadalje, ako Maja želi pronaći željeni a, moramo osigurati redundanciju sustava. Odnosno, moramo zahtijevati $0 \leq a \leq M$, za neku granicu M koja je manja od n. Sada, ako Maja zna indekse "loših" pozicija, i ako umnožak n_i na dobrim pozicijama premašuje M, onda Maja lako može odbaciti pogreške i konstruirati a primjenom Kineskog teorema o ostacima na a_i i n_i koji se nalaze na "dobrim" pozicijama. Međutim, općenito, Maja unaprijed neće znati indekse "loših" pozicija, pa ovaj pristup neće funkcionirati. Unatoč ovim prividnim poteškoćama, Teorem 13. se može iskoristiti za prilično jednostavno rješavanje problema. Neka je P gornja granica umnoška l bilo koji brojeva $n_1, \ldots, n_k$ (primjerice, možemo uzeti da je P umnožak l najvećih brojeva među

$(n_1, \ldots, n_k)$. Nadalje, pretpostavimo da $n > 2MP^2$.

Sada, pretpostavimo da Maja ima neispravne podatke $(b_1, \ldots, b_k)$. U sljedećim koracima navest ćemo što to Maja mora napraviti kako bi pronašla a:

1. Koristeći Kineski teorem o ostacima, dobivamo $b \in \mathbb{Z}$ takav da $0 \leq b < n$ te $b \equiv b_i \pmod{n_i}$, za $i = 1, \ldots, k$.

2. Pokrenuti prošireni Euklidov algoritam na ulazu n, b, dobivamo $EEA(n, b)$, zatim primijeniti Teorem 13. na $n, b, r^* = MP$ i $t^* = P$ kako bismo dobili vrijednosti od r', s', t'.

3. Ako $t' \mid r'$, ispiši $r't'$; inače, ispiši "greška".

Tvrdimo da gornji postupak daje a, pod našom pretpostavkom da skup loših pozicija B duljine najviše l. Kako bismo to dokazali, neka je $t = \prod_{i \in B} n_i$. Prema konstrukciji algoritma, vrijedi $1 \geq t \geq P$. Također, neka je $r = at$ te primijetimo da $0 \leq r \leq r^*$ i $0 < t \leq t^*$. Tvrdimo da vrijedi

$$r \equiv bt \pmod{n}.$$

Kako bismo to doista dokazali, dovoljno je dokazati

$$at \equiv bt \pmod{n_i}$$

za svaki $i = 1, \ldots, k$. Da bismo to pokazali, za svaki indeks i razmatramo dva slučaja:

Slučaj 1: $i \in G$. U ovom slučaju vrijedi $a_i = b_i$, stoga,

$$at \equiv a_i t \equiv b_i t \equiv bt \pmod{n_i}.$$

Slučaj 2: $i \in B$. U ovom slučaju vrijedi $n_i \mid t$, stoga,

$$at \equiv 0 \equiv bt \pmod{n_i}.$$

Prema tome, $at \equiv bt \pmod{n_i}$ vrijedi za svaki $i = 1, \ldots, k$ te vrijedi i $r \equiv bt \pmod{n}$. Nadalje, vrijednosti r', t' dobivene primjenom Teorema 13. zadovoljavaju

$$\frac{r'}{t'} = \frac{r}{t} = \frac{at}{t} = a.$$

Lako se provjeri da je u oba slučaja vrijeme izvršenja jednako $O(len(n)^2)$.

Primjer 11 (vidjeti [3, Primjer 4.6.]). *Pretpostavimo da želimo dešifrirati* 1024*-bitnu poruku zadanu u nizu od* 16*-bitnih blokova, tako da navedena shema može ispraviti najviše* 3 *neispravna bloka. Bez bilo kakvog ispravljanja pogrešaka, mogli bismo to učiniti s* $1024/16 = 64$ *blokova. Međutim, da bismo ispravili ovoliko pogrešaka, potrebno nam je nekoliko dodatnih blokova; zapravo,* 7 *će biti dovoljno. Očigledno,* 1024*-bitna poruka se prirodno može promatrati kao cijeli broj* a *u skupu* $0, \ldots, 2^{1024} - 1$ *te* i*-ti* 16*-bitni blok u kodiranju može se vidjeti kao* a_i *u skupu* $0, \ldots, 2^{16} - 1$. *Postavimo* $k = 71$*, te odabarimo* k *prostih brojeva,* $n_1, \ldots, n_k$*, svaki je duljine* 16*-bitova. Zapravo, izaberimo* $n_1, \ldots, n_k$ *kao najveće proste brojeve manje od* 2^{16}*. Ako to napravimo, dobivamo da je najmanji prost broj među* n_i *upravo* 64717*, koji*

je veći od $2^{15.98}$. *Neka je* $M = 2^{1024}$, *a budući da želimo ispraviti* 3 *pogreške, neka je tada* $P = 2^{3 \cdot 16}$. *Tada zbog* $n = \prod_i n_i$ *slijedi*

$$n > 2^{71 \cdot 15.98} = 2^{1134.58} > 2^{1121} = 2^{1+1024+6 \cdot 16} = 2MP^2.$$

Dakle, s ovim pretpostavkama parametara, gornja shema će ispraviti do 3 *neispravna bloka.*

3.3 Primjene u računalnoj algebri

Racionalna rekonstrukcija također ima brojne primjene u računalnoj algebri. Sada ćemo ukratko skicirati jednu takvu primjenu. Pretpostavimo da želimo pronaći rješenje ν iz jednadžbe

$$\nu A = \omega,$$

gdje nam je kao ulaz dana nesingularna kvadratna cjelobrojna matrica A i cjelobrojni vektor ω. Vektor rješenja ν će općenito imati racionalne elemente. Naglasimo kako želimo izračunati točno rješenje ν, a ne neku njegovu procjenu. Sada, mogli bismo dobiti ν izravno korištenjem Gaussove eliminacije. Međutim, međuvrijednosti izračunate tim algoritmom bile bi racionalne brojevi čiji brojnici i nazivnici mogu postati prilično veliki, što dovodi do prilično dugog procesa računanja. Ipak, može se pokazati da je vrijeme izvršenja polinomijalno. Drugi pristup je računavanje vektora rješenja modulo n, gdje je n potencija prostog broja koji ne dijeli determinantu od matrice A. Pod uvjetom da je n dovoljno velik, tada se vektor rješenja ν može izračunati korištenjem racionalne rekonstrukcije.

4 RSA kriptosustav

Jedna od najzanimljivijih primjena teorije brojeva u posljednjim desetljećima je njezina primjena u kriptografiji. U ovom poglavlju dajemo kratak pregled RSA kriptosustava, nazvan po svojim tvorcima Rivestu, Shamiru i Adlemanu. Nastao je 1977. te je najpoznatiji kriptosustav s javnim ključem. Sigurnost RSA sustava temelji se na problemu faktorizacije velikih prirodnih brojeva. Sada ćemo definirati kriptosustav, a potom i RSA kriptosustav.

Definicija 3. *Kriptosustav je uređena petorka* $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$*, gdje su:*

- P *konačan skup koji sadrži sve moguće osnovne elemente otvorenog teksta;*
- C *konačan skup koji sadrži sve moguće osnovne elemente šifrata;*
- K *konačan skup koji sadrži sve moguće osnovne elemente ključeva;*
- E *skup svih funkcija koji sadrži sve moguće osnovne elemente šifriranja;*
- D *skup svih funkcija koji sadrži sve moguće osnovne elemente dešifriranja;*

pri čemu za svaki $K \in \mathcal{K}$ *postoji funkcija šifriranja* $e_K \in \mathcal{E}$ *i odgovarajuća funkcija dešifriranja* $d_K \in \mathcal{D}$*. Nadalje, funkcije* $e_K : \mathcal{P} \to \mathcal{C}$ *i* $d_K : \mathcal{C} \to \mathcal{P}$ *imaju svojstvo da je* $d_K(e_K(x)) = x$*, za svaki* $x \in \mathcal{P}$*.*

Pogledajmo sada definiciju RSA sustava.

Definicija 4. *Neka je* $n = pq$*, pri čemu su* p *i* q *prosti brojevi. Neka je* $\mathcal{P} = \mathcal{C} = \mathbb{Z}_n = 0, 1, \ldots, n-1$ *te*

$$\mathcal{K} = \{(n, p, q, d, e) : n = pq, de \equiv 1 \pmod{\varphi(n)}\}.$$

Za $K \in \mathcal{K}$ *definiramo*

$$e_K = x^e \pmod{n},$$
$$d_K = y^d \pmod{n}.$$

Vrijednosti n *i* e *su javne, a vrijednosti* p, q *i* d *su tajne, tj.* (n, e) *je javni, a* (p, q, d) *je tajni (privatni) ključ.*

U ovome trenutnu već imamo na raspolaganju koncepte i alate potrebne za razumijevanje ovog kriptosustava, iako ćemo kasnije detaljnije pojasniti pojedine dijelove. Na jednostavnom primjeru pokušat ćemo razjasniti princip RSA kriptosustava.

Pretpostavimo da Ana želi poslati tajnu poruku Maji preko nesigurne mreže. Protivnik može prisluškivati mrežu pa slanje poruke bez šifriranja nije opcija. Korištenje starije, tradicionalnije kriptografije zahtijevalo bi da Ana i Maja međusobno dijele tajni ključ. Međutim, problem nastaje pri stvaranju zajedničkog tajnog ključa i njegove sigurnosti. Kao što smo već naveli, RSA kriptosustav je primjer kriptosustava s javnim ključem. Da bi primijenio

kriptosustav, Maja jednostavno postavlja "javni ključ" u odgovarajući šifrat, dok odgovarajući "privatni ključ" čuva u tajnosti. Kako bi poslala tajnu poruku Maji , Ana dobiva Majin javni ključ i koristi ga za šifriranje svoje poruke. Po primitku šifrirane poruke, Maja koristi Anin privatni ključ za dešifriranje, odnosno, za otkrivanje izvorne poruke.
Evo kako funkcionira RSA kriptosustav. Za generiranje javnog/privatnog ključa, Maja odabire dva vrlo velika, nasumična prosta broja p i q, pri čemu $p \neq q$. Da bismo bili sigurni, p i q bi trebali biti prilično veliki; u praksi se odabire da budu duljine oko 512 bitova. Zatim, Maja računa $n = pq$ te također odabire $e > 1$ tako da vrijedi $gcd(e, \varphi(n)) = 1$. Neka je $\varphi(n) = (p-1)(q-1)$. Naposljetku, Maja računa $d = e^{-1} \mod \varphi(n)$, koristeći prošireni Euklidov algoritam. Javni ključ je definiran s (n, e), a tajni ključ s (n, d). Broj e nazivamo *enkripcijski eksponent*, a d nazivamo *dekripcijski eksponent*. Nakon što Maja objavi svoj javni ključ, Ana može poslati tajnu poruku Maji na sljedeći način.

Pretpostavimo da je poruka kodirana na neki kanonski način kao broj između 0 i $n-1$. Dakle, možemo pretpostaviti da je poruka element α koji dolazi iz $\mathbb{Z}_n$. Za šifriranje poruke, Ana mora izračunati $\beta = \alpha^e$. Tada je šifrirana poruka jednaka β. Kada Maja primi β, ona računa $\gamma = \beta^d$ te interpretira γ kao originalnu poruku.
Najosnovniji zahtjev bilo koje sheme šifriranja je da dešifriranje treba "poništiti" šifriranje. U ovom slučaju, znači da za svaki $\alpha \in \mathbb{Z}_n$, vrijedi

$$(\alpha^e)^d = \alpha.$$

Ako je $\alpha \in \mathbb{Z}_n^*$, onda to očigledno vrijedi, no budući da imamo $ed = 1 + \varphi(n)k$ za neki pozitivan k te uz Eulerov teorem, vrijedi

$$(\alpha^e)^d = \alpha^{ed} = \alpha^{1+\varphi(n)k} = \alpha \cdot \alpha^{\varphi(n)k} = \alpha.$$

Kako bismo pokazali da $(\alpha^e)^d = \alpha$ vrijedi općenito, pretpostavimo da je α proizvoljan element iz $\mathbb{Z}_n$, te pretpostavimo da je $\alpha = [\alpha]_n$. Ako je $\alpha \equiv 0 \pmod{p}$, tada trivijalno slijedi $\alpha^{ed} \equiv 0 \pmod{p}$, inače

$$\alpha^{ed} \equiv \alpha^{1+\varphi(n)k} \equiv \alpha \cdot \alpha^{\varphi(n)k} \equiv \alpha \pmod{p},$$

pri čemu zadnja kongruencija slijedi iz činjenice da je $\varphi(n)k$ višekratnik od $p-1$. Prema tome, pokazali smo da $\alpha^{ed} \equiv \alpha \pmod{p}$. Na istim način možemo dokazati da vrijedi i $\alpha^{ed} \equiv \alpha \pmod{q}$. Dakle, dokazali smo da $(\alpha^e)^d = \alpha$ vrijedi za svaki $\alpha \in \mathbb{Z}_n$.
Naravno, zanimljivo pitanje o RSA kriptosustavu je da li zaista siguran ili nije. Ako je protivnik, s obzirom samo na javni ključ (n, e), mogao faktorizirati n, tada lako može sam izračunati i dekripcijski eksponent d, koristeći isti algoritam koji je koristila Maja. Uvriježeno je mišljenje da je faktorizacija n računski neizvediva, ako je n dovoljno velik, stoga je ovakva metoda napada je neučinkovita, osim ako protivnik nije smislio napredan algoritam faktorizacije. Doista, pokušavati n faktorizirati metodom grube sile je očigledno neizvedivo, ali postoje puno brži algoritmi, no ni oni nisu dovoljno brzi da bi predstavljali ozbiljnu prijetnju sigurnosti RSA kriptosustava.

Može li se razbiti RSA kriptosustav bez faktorizacije n? Na primjer, prirodno je zapitati

se može li se izračunati dekripcijski eksponent d bez faktorizacije od n. Ispostavilo se da je odgovor "ne": kad bi se mogao izračunati dekripcijski eksponent d, tada bi $ed-1$ bio višekratnik od $\varphi(n)$, a može se pokazati kako uz bilo koji dani višekratnik od $\varphi(n)$, lako se može faktorizirati n. Stoga je računanje enkripcijskog eksponenta ekvivalentno faktorizaciji n, pa je ova metoda napada također neučinkovita. Ali još uvijek mogu postojati druge metode napada. Na primjer, čak i ako pretpostavimo da faktoriziranje velikih brojeva je neizvedivo, to nije dovoljno da jamči da za danu šifriranu poruku β, protivnik nije u mogućnosti izračunati β^d (iako nitko zapravo ne zna kako to učiniti bez prethodne faktorizacije n). Doista, jednostavna verzija RSA koja je ovdje predstavljena suočava se s brojnim sigurnosnim problemima (zbog toga su stvarna šifriranja s javnim ključem temeljena na RSA nešto kompliciranije). Jedan takav problem smo već naveli. Pretpostavimo da protivnik koji prisluškuje zna da će Ana poslati jednu od nekoliko poznatih poruka. Primjerice, protivnik može znati da će Ana poslati poruku ili "nađimo se danas" ili "nađimo se sutra". U ovom slučaju, protivnik može šifrirati za sebe svaku od tih poruka , presresti Aninu stvarnu šifriranu poruku, a zatim jednostavnom usporedbom enkripcija, protivnik može odrediti koju je konkretnu poruku Ana šifrirala. Ova metoda napada djeluje jednostavno zato što je algoritam šifriranja deterministički, a zapravo će svaki deterministički algoritam šifriranja biti osjetljiv na ovu metodu napada. Kako bi se izbjegla ova metoda napada, mora se koristiti vjerojatnosni algoritam šifriranja. U slučaju RSA kriptosustava, to se često postiže dodavanjem nekim nasumičnim bitovima prije šifriranja.
Za efikasnost kriptosustava RSA važno je da se modularno potenciranje može efikasno izvoditi. Osnovnu metoda za računanje $e_K(x) = x^e \mod n$, je metoda “kvadriraj i množi” (ovakva metoda se još naziva i “binarne ljestve” jer se u algoritmu rabi binarni zapis eksponenta). Najprije prirodni broj e prikažemo u bazi 2

$$e = 2^{s-1}e_{s-1} + \cdots + 2 \cdot e_1 + e_0,$$

a zatim možemo primijeniti sljedeći algoritam ([3], stranica 99):

$y = 1$
for $i = s-1$ to 0 by -1
  $y = y^2 \mod n$
  if $(e_i = 1)$ then $y = y \cdot x \mod n$.

Očigledno vrijedi da je broj množenja najviše $2s$ pa je ukupan broj operacija $O(\ln e \cdot \ln^2 n)$. Prema tome, zaključujemo da je ovaj algoritam polinomijalan.

Primjer 12. *Uzmimo neka je $p = 3$ i $q = 11$. Slijedi da je $n = 33$ i $\varphi(n) = 20$. Eksponent e mora biti relativno prost s 20 pa neka je $e = 7$. Sada slijedi da je $d = 3$. Imamo naš javni ključ $(n, e) = (33, 7)$. Neka nam sada netko želi poslati poruku $x = 19$. Trebamo izračunati $e_K(x) = 19^7 \mod 33$:*

$$19^7 = 19 \cdot 19^2 \cdot 19^4 = 19 \cdot (-2) \cdot 4 \equiv -20 \equiv 13 \pmod{33}.$$

Dobili smo šifrat $y = e_K(x) = 13$. *Kada dobijemo navedeni šifrat, dešifrirat ćemo ga uporabom tajnog ključa* d*:*

$$x = d_K(y) = 13^3 = 13 \cdot 13^2 = 13 \cdot -4 \equiv -19 \pmod{33}.$$

Dakle, $x = 19$.

Pogledajmo sada jedan primjer RSA kriptosustava nad engleskim alfabetom. Bit će nam potrebna tablica s odgovarajućim brojevima uz slova engleskog alfabeta.

A=1	K=11	U=21
B=2	L=12	V=22
C=3	M=13	W=23
D=4	N=14	X=24
E=5	O=15	Y=25
F=6	P=16	Z=26
G=7	Q=17	
H=8	R=18	
I=9	S=19	
J=10	T=20	

Tablica 2: Engleski alfabet s brojevima.

Primjer 13. *Dešifrirajmo poruku*

$$38 \quad 21 \quad 44 \quad 45 \quad 53 \quad 37$$

u RSA kriptosustavu s javnim ključem $(77, 11)$ *nad engleskim alfabetom.*

Rješenje:
Znamo da je

$$n = 77 = 7 \cdot 11,$$
$$e = 13, \varphi(n) = 60.$$

Sada trebamo pronaći dekripcijski eksponent d. Znamo da za $d, l \in \mathbb{Z}$, vrijedi

$$11d + 60l = 1.$$

Odnosno, vrijedi

$$11d \equiv 1 \pmod{60}.$$

Rješavanjem prethodne kongruencije dobivamo $d = 11$. Dešifrirajmo sada danu poruku s

$d_K(y) = y^{11} \pmod{77}$. Dobivamo

$$\begin{aligned} d_K(38) &= 5 \\ d_K(21) &= 21 \\ d_K(44) &= 11 \\ d_K(45) &= 12 \\ d_K(53) &= 9 \\ d_K(37) &= 4. \end{aligned}$$

Pronađemo odgovarajuća slova i dana poruka glasi EUKLID.

Literatura

[1] A. Dujella, *Teorija brojeva*, Školska knjiga, Zagreb, 2019.

[2] I. Matić, *Uvod u teoriju brojeva*, Odjel za matematiku, Sveučilište J. J. Strossmayera u Osijeku, Osijek, 2015.

[3] V. Shoup, *A Computational Introduction to Number Theory and Algebra*, Cambridge University Press, Cambridge, 2008.

Sažetak

Algoritam je metoda za rješavanje nekog problema. U ovom radu bavit ćemo se Euklidovim algoritmom koji se koristi za pronalaženje najvećeg zajedničkog djelitelja, a iznimnu važnost ima u teoriji brojeva. Osim njega, još ćemo opisati Kineski teorem o ostacima i Fermatov teorem te njihove primjene. Jedna od najzanimljivijih primjena teorije brojeva u posljednjim desetljećima je njezina primjena u kriptografiji. Najpoznatiji kriptosustav s javnim ključem je RSA kriptosustav te ćemo nešto više reći o njemu u radu.

Ključne riječi

djeljivost, Euklidov algoritam, najveći zajednički djelitelj, efikasnost, kriptosustav

Euclidean algorithm

Summary

An algorithm is a step by step method of solving a problem. In this thesis, we will deal with Euclidean algorithm, which is used to find the greatest common divisor, and is extremely important in number theory. In this thesis also describes the Chinese remainder theorem and Fermat's theorem and their applications. One of the more exciting uses of number theory in recent decades is its application to cryptography. The most famous public key cryptosystem is the RSA cryptosystem and we will say something more about it in the thesis.

Keywords

divisibility, the Euclidean algorithm, greatest common divisor, efficiency, cryptosystem

Životopis

Rođena sam 24. lipnja 1998. godine u Požegi. Pohađala sam Osnovnu školu Julija Kempfa u Požegi. Nakon završetka osnovne škole, upisala sam Gimnaziju u Požegi, smjer prirodoslovno-matematička gimnazija. Srednju školu sam završila 2017. te sam iste godine upisala preddiplomskij studij matematike na Odjelu za matematiku u Osijeku. Godine 2020. završavam preddiplomski studij te upisujem diplomski studij, smjer Financijska matematika i statistika.