# Algoritmi u teoriji brojeva

**Antolović, Maja**

**Sveučilište J.J.Strossmayera u Osijeku**
**Odjel za matematiku**
**Preddiplomski studij matematike**

**Maja Antolović**

# Algoritmi u teoriji brojeva

**Završni rad**

**Osijek, 2017.**

**Sveučilište J.J.Strossmayera u Osijeku**
**Odjel za matematiku**
**Preddiplomski studij matematike**

**Maja Antolović**

# Algoritmi u teoriji brojeva

**Završni rad**

Voditelj: izv.prof.dr.sc. Ivan Matić

**Osijek, 2017.**

**Sažetak:** U ovom završnom radu objasnit ćemo što su to algoritmi i navesti primjere. Obradit ćemo Euklidov algoritam i Kineski teorem o ostatcima te navesti algoritme vezane za njega. Objasnit ćemo razvoj u verižni razlomak i navesti algoritme vezane za kvadratne kongruencije.

**Ključne riječi:** Algoritam, Euklidov algoritam, Kineski teorem o ostatcima, verižni razlomak, kvadratne kongruencije.

**Abstract:** In this final work we are going to explain what algorithms are and give examples. We are going to process Euclid's algorithm and Chinese theorem of residues and give algorithms related to him. We are going to explain development of the continued fraction and give algorithms related to congruences involving squares.

**Keywords:** Algorithm, Euclid's algorithm, Chinese theorem of residues, continued fraction, congruences involving squares.

# Sadržaj

# 1 Uvod

Algoritam je metoda za rješavanje neke klase problema, koja za ulazne podatke određenog tipa daje odgovor (izlazne podatke) u konačnom vremenu.
Algoritme ćemo uspoređivati s obzirom na broj "osnovnih koraka" potrebnih za njihovo izvršavanje, te ponekad i s obzirom na potreban prostor (memoriju). Pod osnovnim korakom podrazumjevamo logičku operaciju disjunkcije, konjukcije ili negacije na bitovima - nulama i jedinicama. Veličinu ulaznih podataka ćemo mjeriti brojem bitova potrebnih za njihov prikaz. Osnovna svojstva algoritama su: **definiranost**-uz svaki algoritam moraju jasno biti definirani početni objekti nad kojima se obavljaju operacije, i **konačnost**-algoritam mora biti sastavljen od konačnog broja koraka koji ukazuju na slijed operacija koje treba obaviti nad početnim objektima kako bi se dobili završni objekti ili rezultati.
Obavljanje algoritama naziva se algoritamski proces. Tijekom odvijanja algoritamskog procesa i postupne izgradnje završnog objekta mogu se pojaviti i neki međurezultati.
Za obavljanje algoritma potreban je izvoditelj algoritma, koji razumije algoritam i zna točno obaviti svaki korak algoritma. Trajanje algoritamskog procesa određeno je brzinom kojom izvoditelj obavlja korake algoritma.

**Definicija 1.1.** Polinomijalan algoritam je algoritam čiji je broj operacija u najlošijem slučaju funkcija oblika $O(n^{\mathrm{k}})$, gdje je $n$ duljina ulaznog podatka (u bitovima), a $k$ je konstanta. Algoritme koji nisu polinomijalni, zovemo eksponencijalni.

**Definicija 1.2.** Subeksponencijalni algoritam je algoritam čija je složenost funkcija oblika $O(e^{\mathrm{o(n)}})$, gdje je $n$ duljina ulaznog podatka.

**Definicija 1.3.** Klasa složenosti **P** se sastoji od svih problema odluke za koje postoji polinomijalni algoritam. Klasa složenosti **NP** se sastoji od svih problema odluke za koje se odgovor DA može provjeriti u polinomijalnom vremenu korištenjem neke dodatne informacije, tzv. certifikata. Klasa složenosti **co-NP** se definira na isti način za odgovor NE.

Primjeri algoritama:

**Algoritam za zbrajanje:**
$c = 0$
for $(0 \leq i \leq n)\{$
if $(x_i + y_i + c < b)$ then $w_i = x_i + y_i + c; c = 0$
else $w_i = x_i + y_i + c; c = 1\}$
$w_{n+1} = c$

gdje su $x$ i $y$ dva prirodna broja. Neka su $x = (x_n, ..., x_1, x_0)_b$ i $y = (y_n, ..., y_1, y_0)_b$ dva prirodna broja zapisana u bazi $b$. Tada je $x + y = (w_{n+1}, w_n, ..., w_1, w_0)_b$ u bazi $b$.

**Algoritam za dijeljenje s ostatkom:**
for $(0 \leq i \leq n - t)q_j = 0$
while $(x \geq yb^{n-t})q_{n-t} = q_{n-t} + 1; x = x - yb^{n-t}$
for $(n \geq i \geq t - 1)\{$
if $(x_i = y_t)$ then $q_{i-t-1} = b - 1$
else $q_{i-t-1} = \lfloor (x_i b + x_{i-1})/y_t \rfloor$
while $(q_{i-t-1}(y_t b + y_{t-1}) > x_i b^2 + x_{i-1} b + x_{i-2})$
$q_{i-t-1} = q_{i-t-1} - 1$
$x = x - q_{i-t-1} y b^{i-t-1}$
if $(x < 0)x = x + yb^{i-t-1}; q_{i-t-1} = q_{i-t-1} - 1\}$
$r = x$

**Algoritam za "školsko" množenje:**
for $(0 \leq i \leq n + t + 1)w_i = 0$
for $(0 \leq i \leq t)$ {
$c = 0$
for $(0 \leq i \leq t)$ {
$(uv)_b = w_{i+j} + x_j \cdot y_i + c; w_{i+j} = v; c = u$ }
$w_{n+t+1} = u$ }

Neka su $x$ i $y$ prirodni brojevi i pretpostavimo da je $n \geq t \geq 1$. Želimo naći kvocijent $q = (q_{n-t}, ..., q_0)_b$ i ostatak $r = (r_t, ..., r_0)_b$ pri dijeljenju broja $x$ s $y$, tj. brojeve $q$ i $r$ koji zadovoljavaju $x = qy + r, 0 \leq r < y$.

**Algoritam za računanje** $x \mod m$**:**
$q_0 = \lfloor x/b^n \rfloor; r_0 = x - q_0 b^n; r = r_0; i = 0$
while $(q_i > 0)$ {
$q_{i+1} = \lfloor q_i a/b^n \rfloor; r_{i+1} = q_i a - q_{i+1} b^n;$
$i = i + 1; r = r + r_i$ }
while $(r \geq p)r = r - p$

**Algoritam za računanje $z = x^n$:**
a) Binarna metoda (s desna na lijevo):
$z = 1; y = x$
for $(0 \leq i \leq d-1)$ {
if $(n_i = 1)$ then $z = z \cdot y$
$y = y^2$ }
$z = z \cdot y$

b) Binarna metoda (s lijeva na desno):
$z = x$
for $(d-1 \geq i \geq 0)$ {
$z = z^2$
if $(n_i = 1)$ then $z = x \cdot z$ }

# 2 Euklidov algoritam

**Teorem 2.1.** *Postoje cijeli brojevi x,y takvi da je ax+by=(a,b).*

*Dokaz.* Neka je $g$ najmanji prirodan broj oblika $ax + by, x, y \in \mathbb{Z}$.
Tvrdimo da je $g = (a, b)$. Jasno je da svaki zajednički djelitelj od $a$ i $b$ dijeli $ax + by = g$. Stoga $(a, b)|g$. Pretpostavimo da $g \nmid a$. Tada je $a = qg + r, 0 < r < g$. No, $r = (1 - qx)a - qyb$, pa smo dobili kontradikciju s minimalnošću od $g$. Dakle, $g|a$ i sasvim isto zaključujemo da $g|b$. Stoga je $g \leq (a, b)$, pa zaključujemo da je $g = (a, b)$.

□

Ako su $a$ i $b$ relativno prosti cijeli brojevi, onda postoje cijeli brojevi $x,y$ takvi da je

$$ax + by = 1$$

i pritom je $x \bmod b = a^{-1} \bmod b$, dok je $y \bmod a = b^{-1} \bmod a$.
Euklidov algoritam je jedan od najstarijih, ali ujedno i jedan od najvažnijih algoritama u teoriji brojeva. Zasnovan je na činjenici da je $(a, b) = (b, a \bmod b)$.
Kako je $(a, b) = (|a|, |b|)$, možemo pretpostaviti da $a > b \geq 0$.

**Euklidov algoritam:**
while $(b > 0)(a, b) = (b, a \bmod b)$
return $a$

Da bi analizirali složenost ovog algoritma raspisat ćemo ga po koracima:

$$\begin{aligned} a &= q_1 b + r_1 \\ b &= q_2 r_1 + r_2 \\ &\dots \\ r_{n-3} &= q_{n-1} r_{n-2} + r_{n-1} \\ r_{n-2} &= q_n r_{n-1} \end{aligned}$$

**Primjer 2.1.** *Odredite* $(89, 144)$ *pomoću Euklidovog algoritma.*

*Rješenje:*

$$\begin{aligned} 144 &= 1 \cdot 89 + 55 \\ 89 &= 1 \cdot 55 + 34 \\ 55 &= 1 \cdot 34 + 21 \\ 34 &= 1 \cdot 21 + 13 \\ 21 &= 1 \cdot 13 + 8 \\ 13 &= 1 \cdot 8 + 5 \\ 8 &= 1 \cdot 5 + 3 \\ 5 &= 1 \cdot 3 + 2 \\ 3 &= 1 \cdot 2 + 1 \\ 2 &= 2 \cdot 1 \end{aligned}$$

$(89, 144) = 1$.

Ocijenimo broj koraka, tj. broj $n$, u najlošijem slučaju. Neka su $a$ i $b$ prirodni brojevi takvi da Euklidov algoritam za $(a, b)$ treba $n$ koraka. Tada je $a \geq F_{n+2}, b \geq F_{n+1}$, gdje $F_k$ označava $k$-ti Fibonaccijev broj ($F_0 = 0, F_1 = 1, F_k = F_{k-1} + F_{k+2}$, za $k \geq 2$). Dokažimo to matematičkom indukcijom po $n$. Za $n = 1$ je $b \geq 1 = F_2, a \geq 2 = F_3$. Pretpostavimo da tvrdnja vrijedi za $n$-1 koraka. Za brojeve $b$ i $r_1$ Euklidov algoritam treba $n$-1 koraka. Stoga je po pretpostavci indukcije $b \geq F_{n+1}, r_1 \geq F_n$. No, tada je $a = q_1 b + r_1 \geq b + r_1 \geq F_{n+2}$.
Budući da je $\lfloor F_{k+1}/F_k \rfloor$ i $F_{k+1} \bmod F_k = F_{k-1}$, to su svi kvocijenti u Euklidovom algoritmu za $F_{n+2}$ i $F_{n+1}$ jednaki 1 i algoritam treba točno $n$ koraka. Odatle i iz Binetove formule za Fibonaccijeve brojeve

$$F_n = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right]$$

slijedi

**Teorem 2.2.** *Neka su $a, b \leq N$. Tada je broj koraka u Euklidovom algoritmu za računanje (a,b) manji ili jednak*

$$\left\lceil \frac{\ln(\sqrt{5}N)}{\ln((1+\sqrt{5})/2)} \right\rceil - 2 \approx 2.078 \ln N + 1.672.$$

Euklidov algoritam se može iskoristiti i za nalaženje cijelih brojeva $x$,$y$ takvih da je $ax + by = (a, b)$. Možemo ga koristiti za rješavanje linearnih diofantskih jednadžbi.

**Prošireni Euklidov algoritam:**
$(x, y, g, u, v, w) = (1, 0, a, 0, 1, b)$;
while $(w > 0)\{$
$q = \lfloor q/w \rfloor$;
$(x, y, g, u, v, w) = (u, v, w, x - qu, y - qv, g - qw)$ $\}$
return $(x, y, g)$

Prikazat ćemo još jedan algoritam za računanje najvećeg zajedničkog djelitelja, to je tzv. binarni gcd algoritam. Kod njega se umjesto dijeljenja koriste samo operacije pomaka i oduzimanja. Kao rezultat dobivamo algoritam koji ima veći broj koraka, ali su ti koraci jednostavniji. U samom algoritmu susrećemo dvije ideje. Prva ideja je faktorizacija. Druga ideja je zamjena dijeljenja oduzimanjem, a povezana je s činjenicom da u originalnom Euklidovom algoritmu vrlo često umjesto dijeljenja zapravo imamo oduzimanje, jer je pripadni kvocijent jednak 1.

Označimo s $v_2(k)$ najveću potenciju broja 2 koja dijeli $k$.

**Binarni gcd algoritam:**

$\beta = \min\{v_2(a), v_2(b)\}$
$a = a/2^{v_2(a)}; b = b/2^{v_2(b)}$
while $(a \neq b)$
$(a, b) = (\min\{a, b\}, |b - a|/2^{v_2(b-a)})$
return $2^{\beta}a$

# 3 Kineski teorem o ostatcima

Kineski teorem o ostatcima govori o rješavanju sustava linearnih kongruencija. Ime mu se vezuje uz kineskog matematičara iz prvog stoljeća Sun-Tsua. Smatra se da je taj teorem korišten već u to vrijeme u kineskoj vojsci za prebrojavanje vojnika. Pretpostavimo da treba prebrojiti grupu od približno 1000 vojnika. Vojnici se rasporede npr. u 3, 4, 5 i 7 kolona, te se zabilježi koliko je vojnika ostalo kao "višak" u zadnjem redu. Tako dobivamo sustav od četiri kongruencije s modulima 3, 4, 5 i 7, a taj sustav prema sljedećem teoremu ima jedinstveno rješenje između 800 i 1200.

**Teorem 3.1.** *Neka su $m_1, ..., m_k$ u parovima relativno prosti prirodni brojevi, tj.$(m_i, m_j) = 1$ za $i \neq j$. Tada za proizvoljne cijele brojeve $x_1, ..., x_k$ postoji cijeli broj $x$ takav da vrijedi*

$$x \equiv x_i \pmod{m_i}, i = 1, ..., k.$$

*Broj $x$ je jedinstven modulo $M = m_1 \cdots m_k$.*

Broj $x$ iz teorema možemo naći na sljedeći način. Neka je $M_i = \frac{M}{m_i}$. Kako je $(M_i, m_i) = 1$, to pomoću Euklidovog algoritma možemo naći $a_i$ takav da je $a_i M_i \equiv 1 \pmod{m_i}$. Sada

$$x = \sum_{i=1}^{k} a_i M_i x_i \mod m$$

zadovoljava uvjete teorema.

**Primjer 3.1.** *Riješite sustav kongruencija:* $x \equiv 2 \pmod 3$ $x \equiv 3 \pmod 5$ $x \equiv 2 \pmod 7$.

*Rješenje:* 3, 5, 7 su u parovima relativno prosti pa možemo primijeniti Kineski teorem o ostatcima. $M = 3 \cdot 5 \cdot 7 = 105$.

$M_1 = \frac{M}{m_1} = 35, \quad M_2 = 21, \quad M_3 = 15$

Trebamo rješiti sljedeće kongruencije
$35x_1 \equiv 2 \pmod 3 \Leftrightarrow 2x_1 \equiv 2 \pmod 3 \Rightarrow x_1 = 1$
$21x_2 \equiv 3 \pmod 5 \Leftrightarrow x_2 \equiv 3 \pmod 5 \Rightarrow x_2 = 3$
$15x_3 \equiv 2 \pmod 7 \Leftrightarrow x_3 \equiv 2 \pmod 7 \Rightarrow x_3 = 2$
$x \equiv 35 \cdot 1 + 21 \cdot 3 + 15 \cdot 2 \pmod{105}$
$x \equiv 128 \equiv 23 \pmod{105}$.

Složenost algoritma je $O(\ln^2 M)$.
Kineski teorem o ostatcima ima brojne primjene. Jedan od razloga je da on omogućava da se računanje po jednom velikom modulu zamjeni s nekoliko neovisnih računanja po

puno manjim modulima.
U primjenama su često $m_i$-ovi fiksni, dok $x_i$-ovi variraju. U tom slučaju je dobro onaj dio algoritma koji ne ovisi o $x_i$-ovima izračunati unaprijed. Sljedeći algoritam koristi tu ideju, i vodi računa o racionalnom korištenju brojeva $M_i$ koji mogu biti jako veliki (za razliku od brojeva $a_i$ koji su sigurno manji od $m_i$).

## 3.1 Garnerov algoritam za Kineski teorem o ostatcima

for $(1 \leq i \leq k-1)\{$
$\mu_i = \prod_{j=1}^{i} m_j$;
$c_i = \mu_i^{-1} \mod m_{i+1}$ $\}$
$M = \mu_{k-1} m_k$

$x = x_1$
for $(1 \leq i \leq k-1)$ $\{$
$y = ((x_{i+1} - x)c_i) \mod m_{i+1}$;
$x = x + y\mu_i$ $\}$
$x = x \mod M$

## 3.2 Induktivni algoritam za Kineski teorem o ostatcima

Ukoliko treba rješiti neki jednokratni problem, onda nema koristi od prethodnog računanja s $m_i$-ovima. U takvoj se situaciji preporuča induktivna uporaba originalnog algoritma za sustav od dvije kongruencije. Ako želimo riješiti sustav

$$x \equiv x_1 \pmod{m_1}, \qquad x \equiv x_2 \pmod{m_2},$$

onda jednom primjenom Euklidovog algoritma dobivamo oba željena inverza iz $um_1 + vm_2 = 1$. Tada je $x = um_1x_2 + vm_2x_1 \mod m_1m_2$ rješenje sustava.

**Induktivni algoritam za Kineski teorem o ostatcima:**

$m = m_1; x = x_1$
for $(2 \leq i \leq k)$ $\{$
nađi $u$,$v$ takve da je $um + vm_i = 1$;
$x = umx_i + vm_ix$;
$m = mm_i$;
$x = x \mod m$ $\}$

# 4 Verižni razlomci

Iz prvog koraka u Euklidovom algoritmu imamo

$$\frac{a}{b} = q_1 + \frac{1}{b/r_1}.$$

Drugi korak daje

$$\frac{a}{b} = q_1 + \frac{1}{q_2 + \frac{1}{r_1/r_2}}.$$

Na kraju dobivamo prikaz racionalnog broja $a/b$ u obliku

$$\frac{a}{b} = q_1 + \cfrac{1}{q_2 + \cfrac{1}{q_3 \ddots + \frac{1}{q_n}}}$$

Ovaj se prikaz naziva razvoj broja $a/b$ u jednostavni verižni razlomak. Općenito, za $\alpha \in \mathbb{R}$ se prikaz broja $\alpha$ u obliku

$$\alpha = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \ddots}},$$

gdje je $a_0 \in \mathbb{Z}$, te $a_1, a_2, ... \in \mathbb{N}$, zove razvoj broja u jednostavni verižni razlomak. Verižni razlomak kraće zapisujemo u obliku $[a_0; a_1, a_2, ...]$. Brojevi $a_0, a_1, a_2, ...$ zovu se parcijalni kvocijenti, a definiraju se na sljedeći način:

$$a_0 = \lfloor \alpha \rfloor, \quad \alpha = a_0 + \frac{1}{\alpha_1}, \quad a_1 = \lfloor \alpha_1 \rfloor, \quad \alpha_1 = a_1 + \frac{1}{\alpha_2}, \quad a_2 = \lfloor \alpha_2 \rfloor, ...$$

Postupak se nastavlja sve dok je $a_k \neq \alpha_k$. Razvoj u jednostavni verižni razlomak broja $\alpha$ je konačan ako i samo ako je $\alpha$ racionalan broj.

**Primjer 4.1.** *Razvijmo broj $\frac{41}{47}$ u jednostavni verižni razlomak.*

*Rješenje:*

$$\begin{aligned} 47 &= 41 \cdot 1 + 6 \\ 41 &= 6 \cdot 6 + 5 \\ 6 &= 5 \cdot 1 + 1 \\ 5 &= 1 \cdot 5 \end{aligned}$$

Odavde je $\frac{47}{41} = [1; 6, 1, 5]$, pa je $\frac{41}{47} = [0; 1, 6, 1, 5]$.

**Algoritam za razvoj u verižni razlomak**

$i = 0$
$q = \lfloor a/b \rfloor; r = a - bq; r' = a' - b'q$
while $(0 \leq r' < b'$ and $b \neq 0)\{$
$a_i = q$
$i = i + 1;$
$a = b; b = r; a' = b'; b' = r';$
$q = \lfloor a/b \rfloor; r = a - bq; r' = a' - b'q$ $\}$
if $(b = 0$ and $b' = 0)$ then return $[a_0; a_1, ..., a_i]$
if $(b \neq 0$ and $b' = 0)$ then return $[a_0; a_1, ..., a_{i-1}], a_i \geq q$
$q' = \lfloor a'/b' \rfloor$
if $(b = 0$ and $b' \neq 0)$ then return $[a_o; a_1, ..., a_{i-i}], a_i \geq q'$
if $(bb' \neq 0)$ then return $[a_0; a_1, ..., a_{i-i}], \min\{q, q'\} \leq a_i \leq \max\{q, q'\}$

U slučaju kada je $\alpha$ kvadratna iracionalnost, tj. iracionalan broj koji je rješenje neke kvadratne jednadžbe s racionalnim koeficijentima, tada je njegov razvoj u jednostavni verižni razlomak periodičan. Razvoj se može dobiti sljedećim algoritmom. Prikažemo $\alpha$ u obliku

$$\alpha = \alpha_0 = \frac{s_0 + \sqrt{d}}{t_0},$$

gdje su $d, s_0, t_0 \in \mathbb{Z}, t_0 \neq 0, d$ nije potpun kvadrat i $t_0 \mid (d - s_0^2)$. Zadnji uvjet se uvijek može zadovoljiti množenjem brojnika i nazivnika s prikladnim cijelim brojem. Sada parcijalne kvocijente $a_i$ računamo rekurzivno na sljedeći način:

$$a_i = \lfloor \alpha_i \rfloor, \quad s_{i+1} = a_i t_i - s_i, \quad t_{i+1} = \frac{d - s_{i+1}^2}{t_i}, \quad \alpha_{i+1} = \frac{s_{i+1} + a_0}{t_{i+1}}.$$

Uočimo da iako je $\alpha$ iracionalan broj, ovaj algoritam radi samo s cijelim brojevima. Može se pokazati da su $s_i$-ovi i $t_i$-ovi ograničeni, pa stoga mogu poprimiti samo konačno mnogo vrijednosti. To znači da postoje indeksi $j, k, j < k$, takvi da je $s_j = s_k$ i $t_j = t_k$. No, tada je $\alpha_j = \alpha_k$, što znači da je

$$\alpha = [a_0; a_1, ..., a_{j-1}, \overline{a_j, a_{j+1}, ..., a_{k-1}}],$$

gdje povlaka označava dio koji se periodički ponavlja.
U slučaju kada je $\alpha = \sqrt{d}$ može se preciznije reći kako izgleda razvoj u verižni razlomak. Vrijedi

$$\sqrt{d} = [a_0; \overline{a_1, a_2, ..., a_{r-1}, 2a_0}],$$

gdje je $a_0 = \lfloor \sqrt{d} \rfloor$, a $a_1, ..., a_{r-1}$ su palindromi, tj. $a_i = a_{r-i}$ za $i = 1, 2, ..., r - 1$. Npr. $\sqrt{101} = [10; \overline{20}]$.

Vrijedi da je $r = O(\sqrt{d} \log d)$.

Racionalne brojeve

$$\frac{p_k}{q_k} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \ddots + \cfrac{1}{a_k}}}$$

zovemo konvergente verižnog razlomka. Brojnici i nazivnici konvergenti zadovoljavaju sljedeće rekurzije:

$$p_0 = a_0, \quad p_1 = a_0 a_1 + 1, \quad p_{k+2} = a_{k+2} p_{k+1} + p_k,$$

$$q_0 = 1, \quad q_1 = a_1, \quad q_{k+2} = a_{k+2} q_{k+1} + q_k.$$

Indukcijom se lako dokazuje relacija:

$$q_k p_{k-1} - p_k q_{k-1} = (-1)^k.$$

Relacija $q_k p_{k-1} - p_k q_{k-1} = (-1)^k$ povlači da je $\frac{p_{2k}}{q_{2k}} \leq \alpha$ i $\alpha \leq \frac{p_{2k+1}}{q_{2k+1}}$ za svaki $k$. Nadalje, ako je $\alpha$ iracionalan, onda je $\lim_{k\to\infty} \frac{p_k}{q_k} = \alpha$.

Racionalni brojevi $\frac{p_k}{q_k}$ jako dobro aproksimiraju $\alpha$. Preciznije,

$$\left|\alpha - \frac{p_k}{q_k}\right| < \frac{1}{q_k^2}.$$

Vrijedi i obrat. Ako je $\frac{p}{q}$ racionalan broj koji zadovoljava nejednakost

$$\left|\alpha - \frac{p}{q}\right| < \frac{1}{2q^2},$$

onda je $\frac{p}{q} = \frac{p_k}{q_k}$ za neki $k$.

Konvergente se javljaju i kod rješavanja nekih diofantskih jednadžbi. Posebno je važna njihova uloga u rješavanju Pellovih jednadžbi. To su jednadžbe oblika $x^2 - dy^2 = 1$, gdje je $d$ prirodan broj koji nije potpun kvadrat. Često se uz ovu jednadžbu promatra i jednadžba $x^2 - dy^2 = -1$. Ako sa $\frac{p_k}{q_k}$ označimo konvergente u razvoju u verižni razlomak broja $\sqrt{d}$, onda vrijedi

$$p_k^2 - dq_k^2 = (-1)^{k+1} t_{k+1},$$

gdje je niz $(t_k)$ definiran u algoritmu za razvoj od $\sqrt{d}$. Odavde se lako dobije da jednadžba $x^2 - dy^2 = 1$ uvijek ima (beskonačno) rješenja u prirodnim brojevima, dok jednadžba $x^2 - dy^2 = -1$ ima rješenja ako i samo ako je duljina perioda $r$ u razvoju od $\sqrt{d}$ neparna. Ako je $(X, Y)$ najmanje rješenje u prirodnim brojevima jednadžbe $x^2 - dy^2 = 1$, onda je $(X, Y) = (p_{r-1}, q_{r-1})$ ili $(p_{2r-1}, q_{2r-1})$ u ovisnosti o tome je li duljina perioda $r$ parna ili neparna.

## 4.1 Hermiteova konstrukcija verižnih razlomaka

Neka je $z$ neko rješenje kongruencije $z^2 \equiv -1 \pmod{p}$. Tada je $z^2+1$ neki višekratnik od $p$ kojeg znamo prikazati kao zbroj dva kvadrata. Želimo pomoću njega naći prikaz broja $p$ u tom obliku. Promotrimo verižni razlomak

$$\frac{z}{p} = [a_0; a_1, ..., a_m].$$

Postoji jedinstveni cijeli broj $n$ takav da je $q_n < \sqrt{p} < q_{n+1}$. Budući da $\frac{z}{p}$ leži između susjednih konvergenti $\frac{p_n}{q_n}$ i $\frac{p_{n+1}}{q_{n+1}}$, to je

$$\left| z - \frac{p_n}{q_n} \right| < \left| \frac{p_n}{q_n} - \frac{p_{n+1}}{q_{n+1}} \right| = \frac{1}{q_n q_{n+1}}.$$

Dakle, $\frac{z}{p} = \frac{p_n}{q_n} + \frac{\varepsilon}{q_n q_{n+1}}$, gdje je $|\varepsilon| < 1$. Odavde je $zq_n - pp_n = \frac{\varepsilon p}{q_{n+1}}$, pa je $(zq_n - pp_n)^2 < \frac{p^2}{q_{n+1}^2} < p$. Konačno,

$$(zq_n - pp_n)^2 + q_n^2 \equiv q_n^2(z^2+1) \equiv 0 \pmod{p} \quad i \quad 0 < (zq_n - pp_n)^2 + q_n^2 < 2p,$$

što povlači da je $(zq_n - pp_n)^2 + q_n^2 = p$.

## 4.2 Legendreova konstrukcija verižnih razlomaka

Promotrimo Pellovu jednadžbu $x^2 - py^2 = 1$. Ona ima beskonačno mnogo rješenja u prirodnim brojevima. Neka je $(X, Y)$ najmanje takvo rješenje. Iz $(X+1)(X-1) = pY^2$ slijedi da je $X+1 = ab^2p, X-1 = ac^2$ ili $X+1 = ab^2, X-1 = ac^2p$, gdje je $a = (X+1, X-1) = 1$ ili $2$, $a, b, c$ su neki prirodni brojevi. Odavde je $c^2 - pb^2 = -\frac{2}{a}$ ili $b^2 - pc^2 = \frac{2}{a}$. Zbog minimalnosti od $(X, Y)$, otpada mogućnost $a = 2$ u drugoj jednadžbi. Dakle, imamo

$$c^2 - pb^2 = -1, \quad c^2 - pb^2 = -2, \quad ili \quad b^2 - pc^2 = 2.$$

Ako je $p \equiv 1 \pmod{4}$, onda je $c^2 - pb^2 \equiv 0, 1$ ili $3 \pmod{4}$, pa stoga mora vrijediti $c^2 - pb^2 = -1$ jer su druga i treća jednadžba nemoguće modulo 4. No, nužan i dovoljan uvjet da bi jednadžba $c^2 - pb^2 = -1$ imala rješenja je da period u razvoju u verižni razlomak broja $\sqrt{p}$ bude neparan. Imamo

$$\sqrt{p} = [a_0; \overline{a_1, ..., a_n, a_n, ..., a_1, 2a_0}].$$

Broj $\alpha_{n+1} = \frac{s_{n+1} + \sqrt{p}}{t_{n+1}}$ je čisto periodičan i dio koji se ponavlja s najmanjim periodom je palidroman. Neka je $\alpha'_{n+1} = \frac{s_{n+1} - \sqrt{p}}{t_{n+1}}$ njegov konjugat. Nije teško za vidjeti da je razvoj od $-\frac{1}{\alpha'_{n+1}}$ također čisto periodičan, s time da se parcijalni kvocijenti unutar perioda pojavljuju

u obrnutom redosljedu od onih kod $\alpha_{n+1}$. Kako je periodski dio od $\alpha_{n+1}$ palindroman, zaključujemo da je $-\frac{1}{\alpha'_{n+1}} = \alpha_{n+1}$. Stoga je

$$\alpha'_{n+1}\alpha_{n+1} = \frac{s_{n+1}^2 - p}{t_{n+1}^2} = -1,$$

tj. $p = s_{n+1}^2 + t_{n+1}^2$.

# 5 Kvadratne kongruencije

**Definicija 5.1.** Neka su $a$ i $m$ relativno prosti cijeli brojevi i $m \geq 1$. Kažemo da je $a$ kvadratni ostatak modulo $m$ ako kongruencija $x^2 \equiv a \pmod{m}$ ima rješenja. Ako ova kongruencija nema rješenja, onda kažemo da je $a$ kvadratni neostatak modulo $m$.

**Definicija 5.2.** Neka je $p$ neparan prost broj. Legendreov simbol $\frac{a}{p}$ jednak je 1 ako je $a$ kvadratni ostatak modulo $p$, jednak je -1 ako je $a$ kvadratni neostatak modulo $p$, a jednak je 0 ako je $a \equiv 0 \pmod{p}$.

**Definicija 5.3.** Neka je $m$ neparan prirodan broj i $m = \prod p_i^{\alpha_i}$ njegov rastav na proste faktore, te neka je $a$ proizvoljan cijeli broj. Jacobijev simbol $(\frac{a}{m})$ definira se sa

$$\left(\frac{a}{m}\right) = \prod \left(\frac{a}{p_i}\right)^{\alpha_i},$$

gdje $(\frac{a}{p_i})$ predstavlja Legendreov simbol.

**Osnovna svojstva Jacobijevog simbola:**

1) $a \equiv b \pmod{m} \Rightarrow (\frac{a}{m}) = (\frac{b}{m})$.

2) $(\frac{ab}{m}) = (\frac{a}{m})(\frac{b}{m}), \quad (\frac{a}{mn}) = (\frac{a}{m})(\frac{a}{n})$

3) $(\frac{-1}{m}) = (-1)^{(m-1)/2}, \quad (\frac{2}{m}) = (-1)^{(m^2-1)/8}$

4) $(\frac{m}{n})(\frac{n}{m}) = (-1)^{(m-1)(n-1)/4}$ ako su $m$ i $n$ relativno prosti.

**Algoritam za računanje Jacobijevog simbola** $(\frac{a}{m})$
$a = a \mod m$
$t = 1$
while $(a \neq 0)$ {
while ($a$ paran ) {
$a = a/2$
if $(m \equiv 3, 5 \pmod{8})$ then $t = -t$ }
$(a, m) = (m, a)$
if $(a \equiv m \equiv 3 \pmod{4})$ then $t = -t$
$a = a \mod m$ }
if $(m = 1)$ then return $t$
else return 0

**Primjer 5.1.** *Izračunajmo* $\left(\frac{105}{317}\right)$.

*Rješenje:*

$a = 105 \pmod{317} = 105$
$t = 1$
$a \neq 0, a \equiv 1 \pmod 2$ (neparan)
$(a, m) = (317, 105) \quad (a = 317, m = 105)$
$a \equiv 1 \pmod 4$
$a = 317 \pmod{105} = 2$
$a \neq 0, \quad a \equiv 0 \pmod 2$ (paran)
$a = 2/2 = 1$
$m \equiv 5 \pmod 8 \Rightarrow t = -t = -1$
$a \equiv 1 \pmod 2$ (neparan)
$(a, m) = (105, 1)$
$a \equiv 1 \pmod 4 \Rightarrow t = -t = -(-1) = 1$
$m = 1 \Rightarrow$ return 1

**Propozicija 5.1.** *Ako je* $p \equiv 3 \pmod 4$*, onda je* $x = a^{(p+1)/4}$ *rješenje kongruencije* $x^2 \equiv a \pmod p$*.*

*Dokaz.* Budući da je $a$ kvadratni ostatak modulo $p$, iz Eulerovog kriterija imamo $a^{(p-1)/2} \equiv 1 \pmod p$, pa je

$$x^2 \equiv (a^{(p+1)/2}) \equiv a \cdot a^{(p-1)/2} \equiv a \pmod p.$$

□

Prethodnu propoziciju je moguće modificirati i na preostale proste brojeve, uz poznavanje barem jednog kvadratnog neostatka modulo $p$. Ako je $p \equiv 5 \pmod 8$, onda je broj 2 kvadratni neostatak modulo $p$. Upravo ta činjenica se koristi u sljedećoj propoziciji.

**Propozicija 5.2.** *Ako je* $p \equiv 5 \pmod 8$*, onda je jedan od brojeva* $a^{(p+3)/8}$ *i* $2^{(p-1)/4}a^{(p+3)/8}$ *rješenje kongruencije* $x^2 \equiv a \pmod p$*.*

*Dokaz.* Ako je $p = 8k + 5$, onda je $a^{4k+2} \equiv 1 \pmod p$. Odavde je $a^{2k+1} \equiv \pm 1 \pmod p$, pa je $a^{2k+2} \equiv \pm a \pmod p$. Ako u posljednjoj kongruenciji imamo predznak +, onda je $x = a^{k+1} = a^{(p+3)/8}$ rješenje kongruencije $x^2 \equiv a \pmod p$. Ukoliko imamo predznak -, onda iskoristimo činjenicu da je $(\frac{2}{p}) = -1$. To povlači da je $2^{4k+2} \equiv -1 \pmod p$, pa za $x = 2^{(p-1)/4}a^{(p+3)/8}$ vrijedi

$$x^2 \equiv 2^{4k+2}a^{2k+2} \equiv (-1)(-a) \equiv a \pmod p.$$

□

**Algoritam za određivanje kvadratnog korijena modulo $p$**

$a = a \mod p$
if $(p \equiv 3, 7 \pmod 8)$ then {
$x = a^{(p+1)/4} \mod p$;
return $x$ }
if $(p \equiv 5 \pmod 8)$ then {
$x = a^{(p+3)/8} \mod p$;
$c = x^2 \mod p$;
if $(c \neq a \mod p)$ then $x = x \cdot 2^{(p-1)/4} \mod p$;
return $x$ }

Nađi broj $d \in \{2, 3, ..., p-1\}$ takav da je $(\frac{d}{p}) = -1$
Prikaži $p - 1 = 2^s t, t$ neparan
$A = a^t \mod p$
$D = d^t \mod p$
$m = 0$
for $(0 \leq i \leq s-1)$ {
if $((AD^m)^{2^{s-1-i}} \equiv -1 \pmod p)$ then $m = m + 2^i$ }
$x = a^{(t+1)/2} D^{m/2} \mod p$
return $x$

**Cornacchia-Smithov algoritam**

if $((\frac{-d}{p}) = -1)$ then return *nema rješenja*
$z = \sqrt{-d} \mod p$;
if $(2z < p)$ then $z = p - z$
$(a, b) = (p, z)$
$c = \lfloor \sqrt{p} \rfloor$
$t = p - b^2$
if $(t \neq 0 \pmod d)$ then return *nema rješenja*
if ($t/d$ nije potpun kvadrat) then return *nema rješenja*
return $(b, \sqrt{t/d})$

## 5.1 Kvadrati i kvadratni korijeni

Razmatramo pitanje kako za dani prirodni broj $n$ što efikasnije odrediti da li je $n$ potpun kvadrat ili nije, te ako jest potpun kvadrat, kako izračunati njegov kvadratni korijen. Algoritam za računanje $\lfloor \sqrt{n} \rfloor$ je zapravo varijanta Newtonove iterativne metode za približno računanje korijena jednadžbe.

**Algoritam za $\lfloor\sqrt{n}\rfloor$**

$x = n$
$y = \lfloor(x + \lfloor n/x\rfloor)/2\rfloor$
while $(y < x)$ {
$x = y; y = \lfloor(x + \lfloor n/x\rfloor)/2\rfloor$ }
return $x$

Dokažimo da ovaj algoritam stvarno računa $\lfloor\sqrt{n}\rfloor$. Neka je $q = \lfloor\sqrt{n}\rfloor$. Budući da je $\frac{1}{2}(t + \frac{n}{t}) \geq \sqrt{n}$, za svaki pozitivan broj $t$, imamo da je $x \geq q$ u svim koracima algoritma. U zadnjem koraku imamo $y = \lfloor(x + \lfloor n/x\rfloor)/2\rfloor = \lfloor(x + \frac{n}{x})/2\rfloor \geq x$. Želimo dokazati da je $x = q$. Pretpostavimo suprotno, tj. da je $x \geq q + 1$. Tada je

$$y - x = \Big\lfloor \frac{x + \frac{n}{x}}{2} \Big\rfloor - x = \Big\lfloor \frac{\frac{n}{x} - x}{2} \Big\rfloor = \Big\lfloor \frac{n - x^2}{2x} \Big\rfloor.$$

Iz $x \geq q + 1 > \sqrt{n}$ slijedi $n - x^2 < 0$ i $y - x < 0$, što je kontradikcija.

Složenost ovog algoritma je $O(\ln^3 n)$.

Neka je sada $n$ prirodan broj. Želimo provjeriti je li $n$ potpun kvadrat ili nije. Jedna mogućnost je izračunati $\lfloor\sqrt{n}\rfloor$ i provjeriti je li $q^2 = n$. No, većina prirodnih brojeva nisu kvadrati. Stoga bi bilo dobro neke od njih eliminirati na jednostavniji način. Ideja je iskoristiti činjenicu da ako je $n$ potpun kvadrat, onda je $n$ kvadratni ostatak modulo $m$ za svaki $m$ koji je relativno prost s $n$. Ako je $n$ kvadratni neostatak modulo neki $m$ s kojim je $n$ relativno prost, onda $n$ sigurno nije kvadrat. Ova ideja se u praksi realizira tako da se izabere nekoliko konkretnih modula, te se unaprijed izračunaju kvadrati u pripadnom prstenu.

Za modul $m$, generiramo pripadnu tablicu $qm$ na sljedeći način:
for $(0 \leq k \leq m-1) qm[k] = 0$
for $(0 \leq k \leq \lfloor m/2 \rfloor) qm[k^2 \mod m] = 1$

Jedna preporučena kombinacija modula je 64, 63, 65, 11. Broj kvadrata u $\mathbb{Z}_{64}$, $\mathbb{Z}_{63}$, $\mathbb{Z}_{65}$, $\mathbb{Z}_{11}$ je redom 12, 16, 21, 6. Budući da je

$$\frac{12}{64} \cdot \frac{16}{63} \cdot \frac{21}{65} \cdot \frac{6}{11} = \frac{6}{715} < 0.01,$$

vidimo da na ovaj način za više od 99% brojeva ne moramo računati kvadratni korijen da bi zaključili da nisu kvadrati. Redoslijed kojim testiramo module dolazi od

$$\frac{12}{64} < \frac{16}{63} < \frac{21}{65} < \frac{6}{11},$$

tako da će se za većinu prirodnih brojeva program zaustaviti već nakon prvog testa modulo 64. Mogući su i drugi izbori modula.

**Algoritam za detekciju kvadrata**

$t = n \mod 64$
if $(q64[t] = 0)$ then return $n$ *nije kvadrat*
$r = n \mod 45045$
if $(q63[r \mod 63] = 0)$ then return $n$ *nije kvadrat*
if $(q65[r \mod 65] = 0)$ then return $n$ *nije kvadrat*
if $(q11[r \mod 11] = 0)$ then return $n$ *nije kvadrat*
$q = \lfloor\sqrt{n}\rfloor$ if $(n \neq q^2)$ then return $n$ *nije kvadrat*
else return $n$ *je kvadrat*; $\sqrt{n} = q$

# Literatura

[1] A. Dujella, Teorija brojeva u kriptografiji (skripta), PMF, Matematički odjel, Sveučilište u Zagrebu, 2003/2004.

[2] I. Matić, Uvod u teoriju brojeva, Sveučilište Josipa Jurja Strossmayera u Osijeku - Odjel za matematiku, Osijek, 2015.