# DELJIVOST

Ste že kdaj z manj dela opravili več? Gotovo! Pravijo sicer, da ni kraljevske poti v matematiko in mi se vsekakor ne bomo trudili iskati take. Učenje na lastnih izkušnjah, četudi gre sprva za težave, neuspeh ali celo pravo muko, je v resnici nenadomestljivo. Šele temu običajno sledi večje zadovoljstvo, veselje o odkriti bližnjici. Pravimo tudi, da rešitev pade na rodna tla. V praksi nekatere postopke ponavljamo vedno znova in takrat se pogosto vprašamo, če nam je uspelo najti dovolj učinkovito ali celo optimalno pot do rešitve danega problema. Tisti prvi rešitvi, ki nam je prišla na misel na samem začetku, običajno rečemo

- naivna (če očitno ni optimalna),
- požrešna (če brez premisleka pregleduje),
- izčrpujoča (če pregleda čisto vse možnosti), ...

V tem sestavku bomo razmišljali o učinkovitosti osnovnih računskih operacij.

V šoli nas učijo o deljivosti precej zgodaj. Celo tako zgodaj, da ne poznamo drugega kot desetiški sistem. Tako spoznamo, da se večkratnik števila

- 2 natanko tedaj, ko se končuje na sodo število (v sistemu pri osnovi 2 pa je tako samo število 0),
- 5 natanko tedaj, ko se končuje na 0 ali 5.

Pri deljivosti s 3 (ali 9) ima vsako naravno število enak ostanek kot vsota števk pri osnovi 10 (ki je ravno za ena večja od večkratnika števila 3 oziroma 9). Pri deljivosti z 11 pa ima poljubno naravno število enak ostanek kot razlika vsote števk na sodih mestih in vsote števk na lihih mestih. Opazimo, da smo izpustili praštevilo 7. Osnova 10 mu očitno ni naklonjena, več sreče pa imamo pri osnovi 8. Če privzamemo, da je število 2008 zapisano v binarni obliki, ki je običajna za računalnike, potem izgleda takole:

$$2008 = 2^{10} + 2^9 + 2^8 + 2^7 + 2^6 + 2^4 + 2^3 = 11.111.011.000_2$$

kar se da hitro spremeniti v zapis pri osnovi 8: $3730_8$ (vsaka tri mesta, ki so ločena s pikami ustrezajo eni števki). Če seštejemo vse neničelne števke: $11_2 + 111_2 + 11_2 = 1.101_2$ in to še enkrat ponovimo za dobljeno število: $1_2 + 101_2 = 110_2$, dobimo 6, kar je v resnici ostanek števila 2008 pri deljenju s 7.

Opremljeni s temi pravili in morda še kakšno njihovo kombinacijo (4, 6, 9, 10, 12, 15,...) pričnemo razcepljati števila, a kmalu ugotovimo, da je dobro imeti tabelo praštevil. Naštejmo prvih 100 (ki jih lahko dobimo peš z Erastotenovim rešetom ali pa z enostavim ukazom `seq(ithprime(i),i=1..100)` npr. v MAPLE: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419, 421, 431, 433, 439, 443, 449, 457, 461, 463, 467, 479, 487, 491, 499, 503, 509, 521, 523, 541, ...

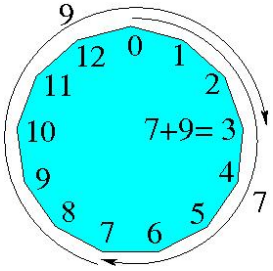
Osnovne oparacije (paralela s konstrukcijami s šestilom in ravnilom):

- zamik (zamaknjene žičke),
- seštevanje (ni samo XOR zaradi problema s prenosom),
- odštevanje (kako ugotovimo katero število je večje),
- množenje (v povprečju $(n/2)$-krat seštevanje in $n$-krat zamik, za dve števili dolžine $n$ v binarnem zapisu),
- deljenje (odštevanje večkratnikov),
- kvadriranje (če smo pri množenju pazljivi, lahko prihranimo skoraj 50%),
- korenjenje (???).

# Kongruence in končni obsegi

V kriptografiji si na splošno radi omislimo končne množice, kot pri številčnici na uri, npr. praštevilske obsege $(\mathbb{Z}_p, +_p, *_p)$, kjer je $\mathbb{Z}_p = \{0, 1, \ldots, p-1\}$.

**Primer:** Naj bo $p = 13$. Namesto s celimi števili računamo z ostanki pri deljenju s 13, tj. z elementi množice $\{0, 1, \ldots, 12\}$. V tem primeru računamo na naslednji način: števili seštejemo ali zmnožimo tako, da običajni rezultat nadomestimo z njegovim ostankom pri deljenju po *modulu* 13. Na primer $7 +_{13} 9 = 7 + 9 \mod 13 = 3$ in $5 *_{13} 4 = 5 * 4 \mod 13 = 7$, saj ima pri deljenju s 13 vsota 16 ostanek 3, produkt 20 pa ostanek 7.




**Kongruence:** naj bosta $a$ in $b$ celi števili in $m$ naravno število.

$$a \equiv b \pmod{m} \iff m \mid b - a.$$

**Red elementa** $g$ v končni (multiplikativni) grupi je najmanjše naravno število $m$, za katerega velja $g^m = 1$.

**Lagrangev izrek:** *V grupi $G$ z $n$ elementi red elementa $g \in G$ deli $n$.*

## Fermatov in Eulerjev izrek

**Fermatov izrek** *Za praštevilo $p$ in $b \in \mathbb{Z}_p$ velja* $b^p \equiv b \pmod{p}$.

### Nekaj napotkov, da Fermatov izrek dokažemo sami

1. Koliko množenj potrebujemo, da izračunamo $m^d$?
2. Prepričaj se, ali je dovolj, da pri RSA uporabimo le Fermatovo kongruenco.
3. Pokaži, da $p \mid \binom{p}{i}$ za $1 < i < p$.
4. Naj bo $p$ praštevilo, potem za poljubni števili $a$ in $b$ velja
$$(a+b)^p \equiv a^p + b^p \pmod{p}.$$
5. Naj bo $p$ praštevilo, potem za poljubno število $m$ velja $m^p \equiv m \pmod{p}$.

**Eulerjevo funkcijo** $\varphi$ definiramo s $\varphi(n) = |\{x \in \mathbb{N} \mid x < n \text{ in } D(x,n) = 1\}|$.

Potem za praštevilo $p$, naravno število $n$ in poljubni tuji si števili $a$ in $b$ velja

$$\varphi(p^n) = p^n - p^{n-1} \quad \text{in} \quad \varphi(ab) = \varphi(a)\varphi(b).$$

Naj bo $p$ praštevilo. Generatorju multiplikativne grupe $\mathbb{Z}_p^*$ pravimo **primitiven element**.

**DN:** Koliko primitivnih elementov ima $\mathbb{Z}_p$?
**DN:** Naj bo $\alpha$ primitiven element, potem za vsak $\beta \in \mathbb{Z}_p^*$ obstaja tak $i \in \{0, 1, \ldots, p-2\}$, da je $\beta = \alpha^i$. Pokaži, da je red elementa $\beta$ enak $(p-1)/D(p-1, i)$.

**Eulerjev izrek** *Če je $a \in \mathbb{Z}_n^*$ oziroma $D(n,a) = 1$, potem velja* $a^{\varphi(n)} \equiv 1 \pmod{n}$.

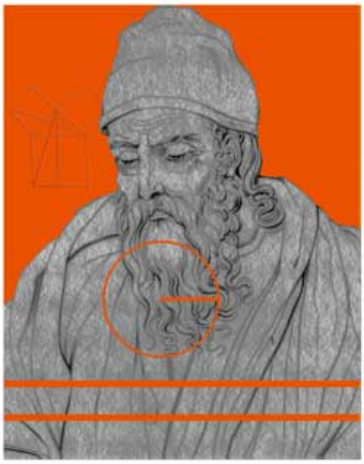
# Zahtevnost Evklidovega algoritma

**Evklidov algoritem** poišče največji skupni delitelj dveh naravnih števil. Zasnovan je na dejstvu, da število $d$, ki deli dve števili, deli tudi njuno razliko. (z drugimi besedami: razlika dveh večkratnikov števila $d$ je prav tako večkratnik števila $d$). V literaturi naletimo nanj prvič leta 300 p.n.š. v 7. knjigi Evklidovih **Elementov**. Nekateri strokovnjaki so mnenja, da je pravi avtor algoritma Eudoxus (l. 375 p.n.š.). Gre za najstarejši (netrivialen) algoritem, ki je preživel do današnjih dni.

Naj bodo $a$, $b$ in $n$ naravna števila, za katera velja $0 \neq b \leq a \leq n$. Pri Evklidovem algoritmu za računanje največjega skupnega delitelja $D(a, b)$ števil $a$ in $b$ najprej delimo $a$ z $b$, oziroma poiščemo natanko določeni števili $s$ in $r$, za kateri velja

$$a = bs + r, \qquad 0 \leq r < b.$$

Če je ostanek $r$ enak 0, potem je $D(a, b) = b$, sicer pa delimo zadnji delitelj z zadnjim ostankom. Postopek ponavljamo, vse dokler ne pridemo do ostanka 0 (to se mora zgoditi, saj se absolutne vrednosti ostankov neprestano zmanjšujejo).



Eukleides
c. 330 - 275 B.C.E.

Potem je zadnji neničeln ostanek največji skupni delitelj števil $a$ in $b$. Ta proces lahko predstavimo z naslednjimi enačbami

$$\begin{aligned}
a &= s_1 b + r_1, & 0 < r_1 < b \\
b &= s_2 r_1 + r_2, & 0 < r_2 < r_1 \\
r_1 &= s_3 r_2 + r_3, & 0 < r_3 < r_2 \\
&\vdots & \\
r_{k-3} &= s_{k-1} r_{k-2} + r_{k-1}, & 0 < r_{k-1} < r_{k-2} \\
r_{k-2} &= s_k r_{k-1} + r_k, & 0 < r_k < r_{k-1} \\
r_{k-1} &= s_{k+1} r_k + 0. &
\end{aligned}$$

Potem je $D(a, b) = r_k$. Definirajmo še $r_{-1} := a$ in $r_0 := b$. Zaporedje $\{r_i\}$ imenujemo *Evklidovo zaporedje* števil $a$ in $b$.

Pri algoritmu je dobro poznati njegovo časovno in prostorsko zahtevnost, ki jo lahko zagrizeni bralci spoznajo skozi naslednje naloge:

(a) Dokaži, da je $r_{i+2} < \frac{1}{2} r_i$ za vsak $1 \leq i \leq k-2$, kadar si za ostanke izbiramo nenegativna cela števila.

(b) Naj bosta $a$ in $b$ naravni števili, $a \geq b$. Dokaži, da je časovna zahtevnost običajnega deljenja velikih števil

$$a = sb + r, \quad 0 \leq r < b$$

$c_1(\log_2 b)(\log_2 s)$ bitnih operacij, kjer je $c_1$ neka od števil $a$ in $b$ neodvisna konstanta.

(c) Iz (a) izpelji, da je časovna zahtevnost Evklidovega algoritma $c_2(\log_2 n)^3$ bitnih operacij, kjer je $c_2$ neka od števil $a$ in $b$ neodvisna konstanta.

(d) Dokaži, da velja $s_1 s_2 \cdots s_{k+1} \leq a$.

(e) Dokaži, da je časovna zahtevnost Evklidovega algoritma $c_3(\log_2 n)^2$ bitnih operacij, kjer je $c_3$ neka od števil $a$ in $b$ neodvisna konstanta (to je seveda boljša ocena kot v (b)).

## Linearna diofantska enačba

Poiščimo celoštevilčne rešitve $(x, y)$ enačbe

$$ax + by \;=\; c \qquad \text{za} \;\; a, b, c \in \mathbb{Z}. \tag{1}$$

Morda boste rekli, da že poznate to enačbo in veste, da ležijo vse takšne rešitve na neki premici. Vendar pa bodite pozorni na zahtevo, da iščemo celoštevilske rešitve. Prav zato rečemo, da gre za diofantsko enačbo (oziroma v našem primeru splošno linearno diofantsko enačbo). Naj bo $d$ največji skupni delitelj števil $a$ in $b$. Če število $d$ ne deli števila $c$ na desni strani enačbe (1), potem očitno ni nobene rešite. Pokazali bomo, da ima v primeru, ko $d$ deli $c$, diofantska enačba (1) vedno rešitve. Opisali bomo postopek za iskanje le-teh.

Naj bo $(x_0, y_0)$ ena rešitev dane enačbe, tj. $ax_0 + by_0 = c$. To enakost odštejemo od dane enačbe (1) in dobimo

$$a\,(x - x_0) + b\,(y - y_0) = 0 \qquad \text{oziroma} \qquad \frac{a}{d}\,(x - x_0) = \frac{b}{d}\,(y_0 - y). \tag{2}$$

Od tod in iz definicije števila $d$ sledi, da kvocient $a/d$, ki predstavlja celo število, deli razliko $y_0 - y$, oziroma da velja $y_0 - y = t\,a/d$ za neko celo število $t$. To vstavimo v dano enačbo in dobimo $x - x_0 = t\,b/d$. Reševanje diofantske enačbe (1) smo prevedli na iskanje ene rešitve, saj smo poljubno rešitev $(x, y)$ enačbe (2) izrazili z eno njeno rešitvijo:

$$x = x_0 + \frac{b}{d}\,t, \qquad y = y_0 - \frac{a}{d}\,t, \qquad\qquad t \in \mathbb{Z}. \tag{3}$$

Kadar smo pred težko nalogo, so nam pogosto v pomoč posebni primeri. V primeru, ko je $c = a$, je ena rešitev diofantske enačbe (1) par $(1, 0)$, v primeru, ko je $c = b$, pa je ena rešitev diofantske enačbe (1) par $(0, 1)$.

Sedaj opišimo postopek za iskanje posebne rešitve diofantske enačbe $ax + by = d$. Dobljeno posebno rešitev nato samo še pomnožimo s številom $c/d$ in že imamo posebno rešitev splošne linearne diofantske enačbe (1). Povežimo primera, ko je desna stran diofantske enačbe (1) enaka enemu izmed števil $a$ in $b$, z zgornjim primerom, ko smo si za desno stran izbrali $d$. Števila $a$, $b$ in $d$ povezuje Evklidov algoritem, zato poskusimo poiskati zaporedji števil $\{p_i\}$ in $\{q_i\}$, za katera velja:

$$a\,p_i + b\,q_i \;=\; r_i, \tag{4}$$

kjer je $\{r_i\}$ Evklidovo zaporedje števil $a$ in $b$, tj. $r_{-1} = a$ in $r_{i+1} = r_{i-1} - s_i\,r_i$, kjer ponavljamo postopek vse dokler ostanek $r_{i+1}$ ne postane enak nič. Zaradi začetnih pogojev si lahko izberemo kar $p_{-1} = 1$, $q_{-1} = 0$ in $p_0 = 0$, $q_0 = 1$. Iz rekurzivne zveze pa z množenjem in odštevanjem treh zaporednih enačb (4) za $i = j - 1, j, j + 1$ dobimo

$$a\,(p_{j-1} - s_j\,p_j) + b\,(q_{j-1} - s_j\,q_j) \;=\; a\,p_{j+1} + b\,q_{j+1},$$

tako da si lahko izberemo še

$$p_{j+1} = p_{j-1} - s_j\,p_j \qquad \text{in} \qquad q_{j+1} = q_{j-1} - s_j\,q_j. \tag{5}$$

Naj bo $k$ tako naravno število, da je $r_{k+1} = 0$. Potem je par $(p_k, q_k)$ rešitev, ki smo jo iskali, zgornje iteriranje skupaj z Evklidovim algoritmom pa je poznano pod imenom *razširjeni Evklidov algoritem.*

*Aleksandar Jurišić*