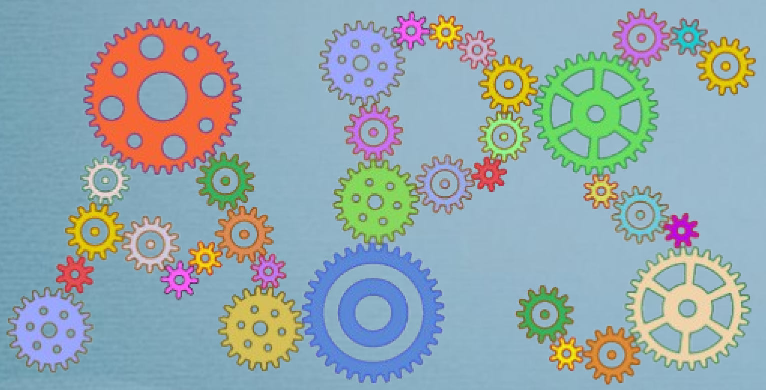
Algoritmi in podatkovne strukture 1

Visokošolski strokovni študij Računalništvo in informatika

Največji skupni delitelj (gcd)



Jurij Mihelič, UniLj, FRI

Največji skupni delitelj

- Algoritem s faktorizacijo
  - faktoriziramo a in b
  - zmnožimo skupne faktorje
  - težava: ne znamo hitro faktorizirati

$15525 = 3^3 \cdot 5^2 \cdot 23$
$2277 = 3^2 \cdot 11 \cdot 23$

$3^2 \cdot 23 = \mathbf{207}$

Največji skupni delitelj

- Zaporedno pregledovanje
  - preizkušamo deljivost
    od večjega proti manjšemu

```
for d = min(a, b) to 1 do
    if d | a and d | b then
        return d
```

$O(n^2 2^n)$

Največji skupni delitelj

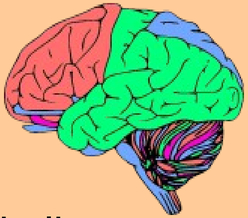
- Za $a, b > 0$ in $a \geq b$, velja
  - $\gcd(a, b) = \gcd(a - b, b)$
  - in posledično še
  - $\gcd(a, b) = \gcd(a \bmod b, b)$

Dokaz za $\gcd(a, b) = \gcd(a - b, b)$

- Če $x \leq y$ in $x \geq y$, potem velja $x = y$

Naj bo d nek skupni delitelj a in b, kar zapišemo kot $d|a$ in $d|b$.

- Nadalje zapišimo $a = d \cdot a'$ in $b = d \cdot b'$ ter
  - posledično $a - b = d(a' - b')$, torej $d|a-b$.
- Drugače povedano d deli tudi razliko $a-b$,
  - torej je d tudi skupni delitelj $a-b$ in b.
- Sedaj vzemimo kar $d = \gcd(a, b)$,
  - ki je delitelj, vendar morda ni največji skupni delitelj,
  - zato lahko zapišemo le $\gcd(a, b) \leq \gcd(a-b, b)$



Na podoben način velja, če $d|a-b$ in $d|b$, potem tudi $d|a$,

- torej $\gcd(a-b, b) \leq \gcd(a, b)$

Največji skupni delitelj

- Evklidov algoritem
  - Neposredno uporabimo izrek
  - $\mathrm{gcd}(a, b) = \mathrm{gcd}(a \bmod b, b)$

Pravilnost
Evklidovega algoritma
sledi iz izreka.

```
fun gcd(a, b) is
   if b == 0 then return a
   return gcd(b, a % b)
```

Največji skupni delitelj

- Evklidov algoritem
  - Če $a \geq b$, potem $a \bmod b < a/2$

    Dokaz (dva primera)
    - $b \leq a/2 \Rightarrow \boldsymbol{a} \textbf{ mod } \boldsymbol{b} < b \leq \boldsymbol{a/2}$
    - $b > a/2 \Rightarrow \boldsymbol{a} \textbf{ mod } \boldsymbol{b} = a - b < a - a/2 = \mathbf{a/2}$

  - Zahtevnost
    - v dveh korakih se a in b vsaj prepolovita
    - potrebujemo torej kvečjemu $2n$ korakov
    - na vsakem koraku eno deljenje $O(n^2)$
    - torej $O(n^3)$
  - Kakšna je zahtevnost,
    - če imamo opravka z malimi števili?

Velikost števila se na vsakem koraku zmanjša za 1 bit.

0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 17711 28657 46368 …

Fibonaccijeva števila

- … in plojenje nesmrtnih zajcev.

- Rekurenčna definicija
  - $F_0 = 0$
  - $F_1 = 1$
  - $F_n = F_{n\text{-}2} + F_{n\text{-}1}$

Leonardo Fibonacci, 1170-1250

Fibonaccijeva števila

- Zlati rez
  - razmerje med dvemi zaporednimi Fibonaccijevimi števili

$$\varphi = \lim_{n \to \infty} \frac{F_{n+1}}{F_n} = \frac{1+\sqrt{5}}{2} \approx 1.618$$

  - hitrost naraščanja

$$F_n \geq \varphi^{n-2}$$

Gabriel Lamé

Največji skupni delitelj

- Evklidov algoritem
  - Št. korakov oz. globina rekurzije $R(a, b)$
  - Najmanjša a in b, kjer $a > b > 0$ in $R(a, b) = N$, sta $a = F_{N+2}$ in $b = F_{N+1}$

```
fun gcd(a, b) is
   if b == 0 then return a
   return gcd(b, a % b)
```

N	q	a	b
9	1	...	
8	1	55	34
7	1	34	21
6	1	21	13
5	1	13	8
4	1	8	5
3	1	5	3
2	1	3	2
1	1	2	1
0	*2*	*1*	*0*

Največji skupni delitelj

- Evklidov algoritem
  - Št. korakov oz. globina rekurzije $R(a, b)$
  - $R(a, b) \leq 5 \cdot \log_{10} b$

```
fun gcd(a, b) is
   if b == 0 then return a
   return gcd(b, a % b)
```

Povzetek

- Največji skupni delitelj – gcd
  - faktorizacija
  - zaporedno preverjanje
  - Evklidov algoritem
- Lastnosti
  - pravilnost sledi iz $\mathrm{gcd}(a, b) = \mathrm{gcd}(b, a \bmod b)$
  - zahtevnost: $O(n^3)$
- Najslabši primer
  - zaporedna Fibonaccijeva števila
  - št. korakov: $\leq 5 \cdot$št. mest v desetiškem zapisu