Algoritmi:

- evklidov
- seštevanje
- urejanje
- simpleks algoritem za linearno programiranje
- algoritmi za preverjanje praštevilnosti
- Shorov kvantni algoritem za faktorizacijo

Ključne lastnosti algoritmov

- netočnost
- končnost
- vhod, izhod
- učinkovitost

Ko imamo algoritem se lahko vprašamo

1) če je pravilen
2) časovna in prostorska zahtevnost
3) Ali je optimalen

Fibonaccijeva števila

$F_0=0 \quad F_1=1$

$F_n = F_{n-1} + F_{n-2}$

FUNCTION rFib(n)

if n=0 : return 0 $\quad 1$

if n=1 : return 1 $\quad 1$

return : $\underbrace{rFib(n-1)}_{1+T(n-1)} + \underbrace{rFib(n-2)}_{1+T(n-2)}$ $\quad 3+T(n-1)+T(n-2)$

$\underbrace{\qquad\qquad}_{1}$

Časovna zahtevnost?

$T(n) = ?$ $\qquad T(0) = 1$

$T(1) = 2$

$T(n) = T(n-1)+T(n-2)+3 \gtrsim F_n = \frac{\varphi^n + \psi^n}{\sqrt{5}}$

$T(n) > \frac{\varphi^n}{\sqrt{5}} \sim \Omega(1{,}6^n)$

$\varphi = \frac{1+\sqrt{5}}{2}$

$\psi = \frac{1-\sqrt{5}}{2}$

INTERaktivni algoritem

Function iFib(n)

if n=0 : return 0

if n=1 return 1

create an array f[0....n]

f[0]=0 f[1]=1

FOR i=2....n

f[i]=f[i-1]+f[i-2]

return f[n]

Do 16. strani v skripti

Deli in Vladaj

Razbijemo problem na manjše podprobleme

1) Razbijemo
2) rekurzivno rešimo
3) združi rešitve

$$
\begin{array}{r}
1011 \\
\times\ 1101 \\
\hline
1011 \\
0000 \\
1011 \\
1011 \\
\hline
\ldots
\end{array}
\quad \Big\}\ n \text{ korakov}
$$

$T(n) = O(n^2$

$$x = [x_L \mid x_D] = 2^{\frac{n}{2}} x_L + x_D$$

$$y = [y_L \mid y_D] = 2^{\frac{n}{2}} y_L + y_D$$

$$xy = (2^{\frac{n}{2}} x_L + x_D)(2^{\frac{n}{2}} y_L + y_D) =$$

$$2^n x_L y_L + 2^{\frac{n}{2}} (x_L y_D + y_L x_D) + x_D y_D)$$

$a = 4$

$b = 2$ $\quad 4 > 2 \Rightarrow O(n^{\log_2 4}) = O(n^2)$

$d = 1$

kerratauban algoritem