Def. Θ-notacija

$f(n) = \Theta\, g(n) \iff \exists\; c_1, c_2, n_0 : c_1 |g(n)| \le |f(n)| \le c_2 |g(n)|$

$\forall n \ge n_0$

ili:
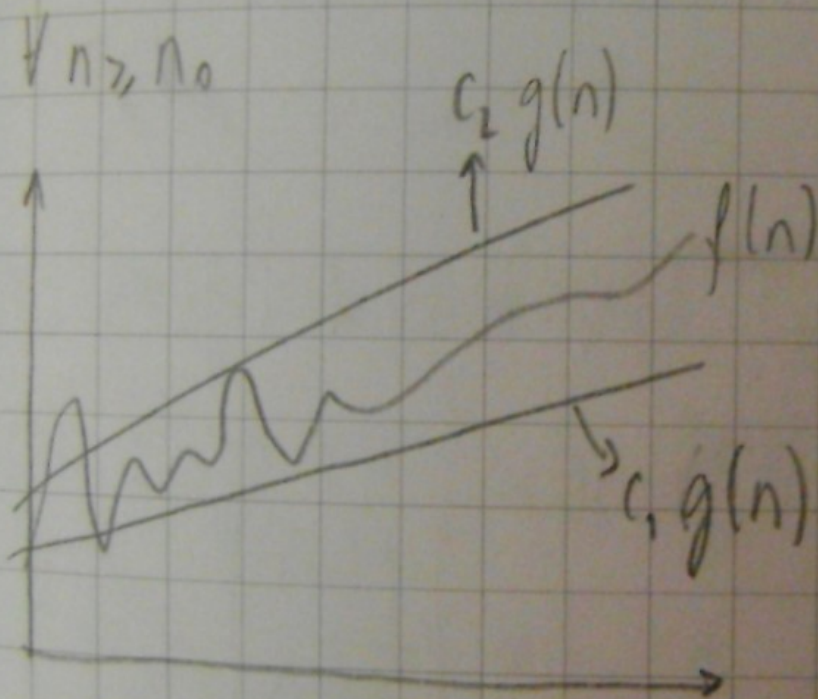
$$0 < \lim_{n \to \infty} \left| \frac{f(n)}{g(n)} \right| < \infty$$

asimptotska čvrsta ograda

$c_2 g(n)$

$f(n)$

$c_1 g(n)$

Pr.) $f_1(n) = n^2 + 3n + 4$

$f_1(n) = O(n^2) = \Theta(n^2)$

$f_2(n) = 3n + 4$

$f_2(n) = O(n)$ ali $f_2(n) = n^2$
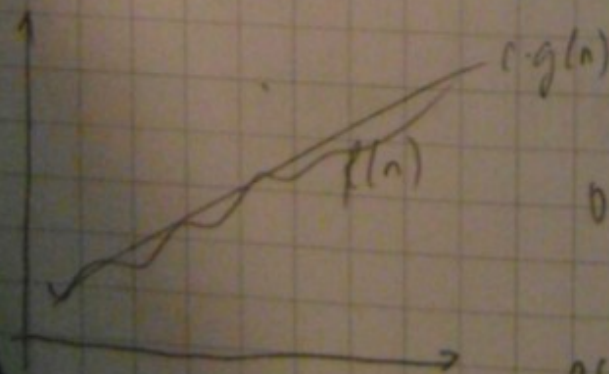
$f_2(n) = \Theta(n)$

$f_2(n) \neq \Theta(n^2)$

napomena: za veliko O notaciju u zadacima
traži se najbolja moguća gornja ograda
koja je jednaka Θ.

OCJENA REDA SLOŽENOSTI

DEF. O-notacija

$f(n) = O(g(n)) \Leftrightarrow \exists\, c, n_0 : |f(n)| \leq c \cdot |g(n)|\ \forall n \geq n_0$

$c \cdot g(n)$

$f(n)$

ili :

$0 \leq \lim\limits_{n \to \infty} \left| \dfrac{f(n)}{g(n)} \right| < \infty$

asimptotska gornja ograda

zad. $x = q = z = 1$ $k = 0$

za (i=1 do 100)

$\llcorner$ $x\ /= 0.55;$

za (i=1 DO 2n)

$\quad$ $y *= X\ / i;$

$\quad$ $k\ += i;$

$\quad$ $j = k;$

$\quad$ ponavljaj

$\quad\quad$ $z *= y / k;$

$\quad\quad$ $j--;$

$\quad$ dok je $(j > 0)$

Odredite br. operacija i složenost algoritma uz $*$ i $/$ kao jediničnu mjeru.

$$\sum_{i=1}^{n} i^2 = \frac{n}{6}\left(2n^2 + 3n + 1\right)$$

$f(n) = 100 + 2n \cdot 2 + h(n)$

$h(n) = \sum_{i=1}^{2n} \sum_{j=1}^{k} 2 = \left| k(i) = \frac{i(i+1)}{2} \right| = 2 \sum_{i=1}^{2n} \sum_{j=1}^{i(i+1)/2} 1 =$

$= \not{2} \sum_{i=1}^{2n} \frac{i(i+1)}{\not{2}} = \sum_{i=1}^{2n} i^2 + \sum_{i=1}^{2n} i = \frac{2n}{6}\left(2(2n)^2 + 3 \cdot 2n + 1\right) +$

$\frac{2n(2n+1)}{2} = \frac{8}{3}n^3 + 4n^2 + \frac{4}{3}n$

$f(n) = O(n^3)$

Zad.) početak (n)

$\quad$ i=1;

$\quad$ k=n-1;

$\quad$ fja(n,k,i);

fja(n,k,i)

$\quad$ k=k-i;

$\quad$ i=i*2;

$\quad$ ako (k>0)

$\qquad$ fja(n,k,i)

$\quad$ for (i=1 do n)

$\qquad$ k=(k*3)/(i*2);

BRPF (br. pozivanja f-je)

$1+2+4+8 \ldots \le k$

članova = BRPF $\qquad$ BRPF = $\log_2 n$

$\sum\limits_{i=0}^{BRPF-1} 2^i = k$

$2^{BRPF} - 1 = k = n-1$

$2^{BRPF} = n$

- unutar f-je: $1+3n$

$f(n) = \log(n)(1+3n) = 3n\log n + \log n$

$f(n) = o(n\log n)$

zad)   početak (n)

$t = n\cdot(n+1)/2;$

$i = j = 1;$

dok je $(i < t)$

radinesto();

$j++;$

$i = i + j;$

Algoritam radinesto() je složenosti $o(n\log n)$

Odredite složenost svega.

Brojimo samo br. pozivanja (BP) f-je.

$i = 1 + 2 + 3 + \dots = \sum_1^{BP} z = \frac{BP(BP+1)}{2} = t = \frac{n\cdot(n+1)}{2} \longrightarrow BP = n$

$f(n) = n\cdot n\log n = n^2\log n = o(n^2\log n)$

zad.)   poč(n)

$i=1; k=n^2;$

$\{j^a (n,k,i);$

$\{j^a (n,k,i)$

$k = k-i;$

$i = i+1;$

ako $(k>0)$

$\{j^a (n,k,i);$

// $2n$ operacija

BRPF = F

$1+2+3+\dots+F = k = n^2$

$$\frac{F(F+1)}{2} = n^2$$

$F^2 + F - 2n^2 = 0$

$F = -\frac{1}{2} \pm \frac{2\sqrt{2}n}{2} \approx n\sqrt{2}$

$\{(n) = F \cdot 2n = n^2 \sqrt{2} = O(n^2)$