

Multími

- Reflexivā

$$a R a$$
$$\nexists a \in S$$

- Simetricā

$$a R b \Rightarrow b R a$$

- Antisim.

- Transitivā

ceil \nearrow floor \searrow
[] approx. []

Sume Σ Recurrente

Gauss ; $\sum_{i=1}^n i^2 = \frac{2n^3 + 3n^2 + n}{6}$

$$\text{factorial : } \left\{ \begin{array}{l} n! = (n-1)! \cdot n, n \geq 1 \\ 1! = 0! = 1 \end{array} \right.$$

$$\text{Fibonacci : } \left\{ \begin{array}{l} f(n) = f(n-1) + f(n-2) \\ f(1) = f(0) = 1 \end{array} \right.$$

```

unsigned long factorial( uint n ) {
    if ( n == 0 )
        return 1;
    else
        return n * fact( n - 1 );
}

```

$$T(0) = C \quad (\text{const. timp})$$

$$T(1) = C + T(0) = 2C$$

$$T(2) = C + \underline{T(1)} = 3C$$

$$T(n) = (n+1)C$$

Tehnici demonstrare

- Dem. directe (deducție)
- Contradicție
- Inducție

$p, p, a \models \exists \text{ lui } \text{Max}$

dacă $\text{Max} \in \mathbb{N} \rightarrow \text{Max} + 1 \notin \mathbb{N}$

$\text{Max} + 1 > \text{Max} \Rightarrow \nexists \text{ lui}$

Inductia

- Cas initial $P(k_0)$ adeu.
- Pasul de inductie $P(k) \Rightarrow P(k+1)$

Caz de baza \rightarrow stop. la rec.

if ($n = 0$)
return 0

Rez de ind. \rightarrow rez. $n \neq f(n-1)$

Interval (Rec)

```
void f ( int start, int end ) {
```

if (start == end) cas
baza

else

$\overbrace{f(start, end - 1)}$ pas
ind.

Algoritmi

→ Metoda pt. a obține un rezultat

Propri. → G.

→

→

Tim. Ruleș / Mem. Consumată

Analiza Algoritmilor

for $c \cdot n \cdot n$

for $T(n) = c n^2$

3 caiuri

- }) Cel mai favorabil
- } Cel mai nefavorabil
- { Average case

Notări asimptotice

fct. $g(n) \rightarrow \Theta(g(n))$ (theta)

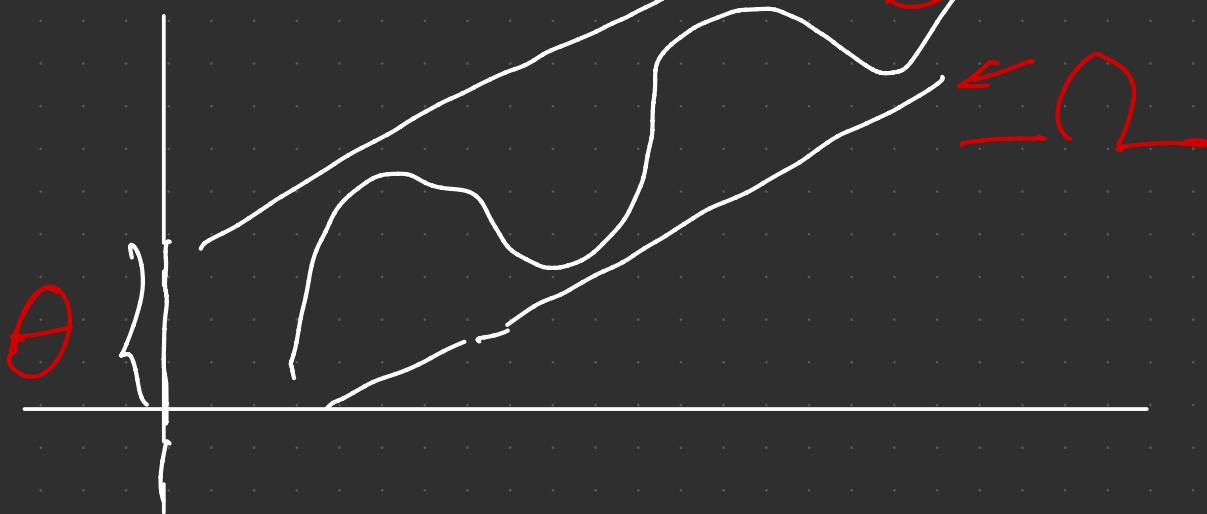
$\Theta(g(n)) = \{ f(n); \exists \text{ const} > 0$
 $c_1, c_2 \text{ și } n_0 \text{ a.} \}$

$$0 \leq c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$$

$\forall n \geq n_0$

$$\frac{1}{2} n^2 - 3n = \Theta(n^2)$$

$$C_1 \cdot n^2 \leq \frac{1}{2} n^2 - 3n \leq C_2 n^2$$



Aprezarea + execuție

p.t. + constant $\Theta(1) \Rightarrow O(1)$
 $(O(n^0))$

Nu \exists alte constante în $O(\dots)$

for (...) { mc
 a[] ... }

for($i = 0; i < 100; i++$)

$$a[i] = i + 1$$

$$C \cdot 100 = C \rightarrow O(1)$$

for($i = 0; i < n; i++$)

 for($j = 0, j < i, j++$) $\{ O(n^2)$

$$a[ij] = j + 1$$

$$C + C + \dots + C = nc$$

n

for($k = 1; k < n; k \times 2$)

~~log n ori~~

Căutarea binară

$T(n) \approx 1$

$$T(n) = T(n/2) + 1$$

$$T(n) = \log n \Rightarrow \Theta(n)$$

Profiling Algoritmos

→ VS Profiling (CPU, Mem)