



UNIVERSITATEA BABEȘ-BOLYAI
Facultatea de Matematică și Informatică



ALGORITMI ȘI PROGRAMARE

Cursul 10

Complexități

Sortări

Ionescu Vlad

vlad.ionescu@ubbcluj.ro

Feedback săptămâna 9

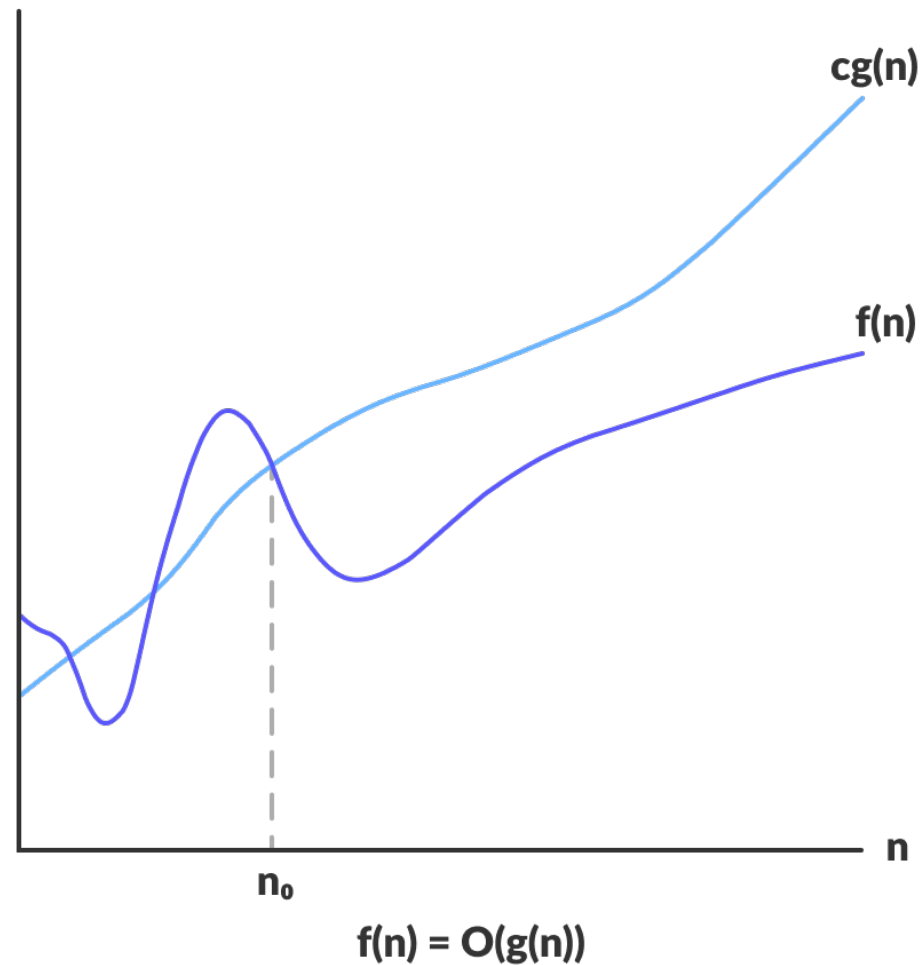
- Întrebări
- Feedback

Complexități

- Notațiile asimptotice
- Sunt **proprietăți** ale timpului de execuție sau ale memoriei folosite
- Prin memorie folosită, înțelegem **memoria suplimentară**, adică ignorăm datele de intrare în calcul

Complexități – notația O (big oh)

- Reprezintă o limită superioară



Complexități – notația O

▣ Definiție:

$$f(n) \in O(g(n)) \Leftrightarrow \exists c \in \mathbb{R}^+, n_0 \in \mathbb{R} \text{ s.t. :}$$

$$|f(n)| \leq cg(n) \forall n \geq n_0$$

$$\lim_{n \rightarrow \infty} \sup \frac{|f(n)|}{g(n)} < \infty$$

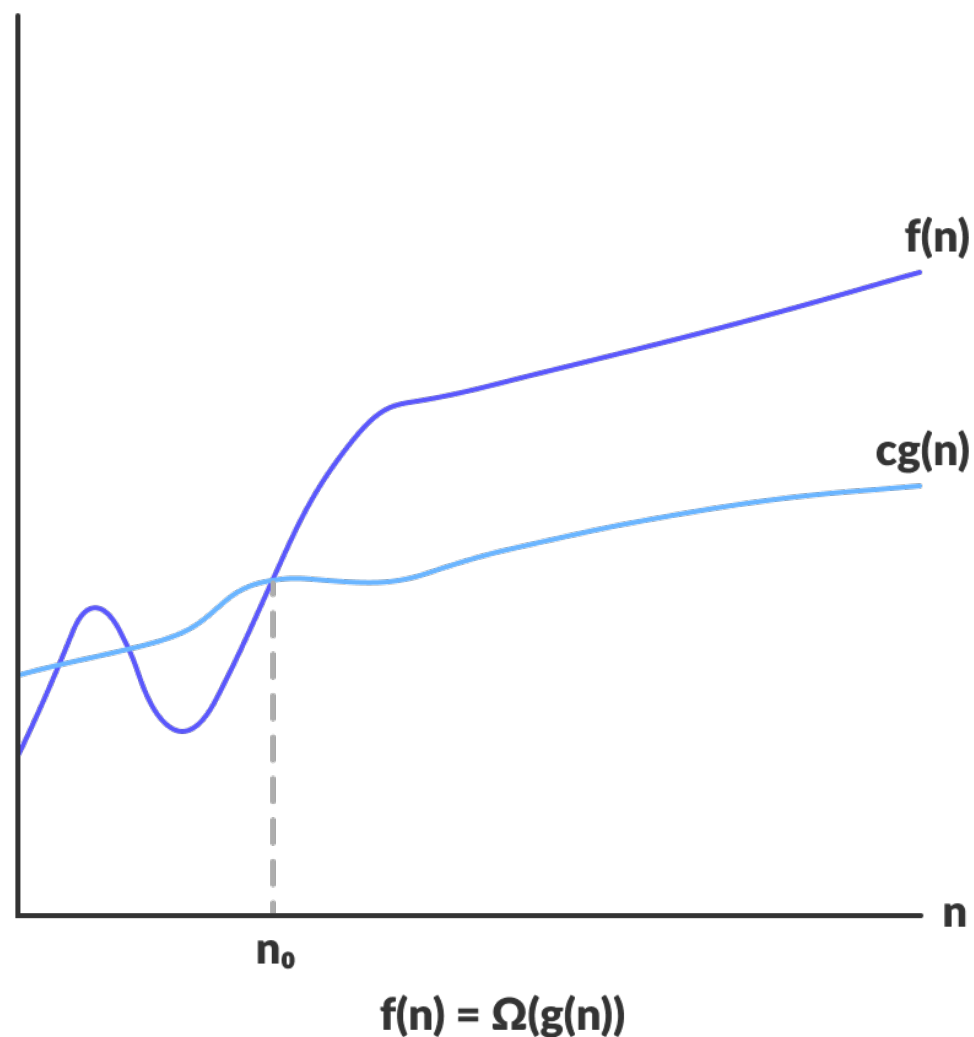
Complexități – notația O

□ Example:

- $f(n) = n^2 + 2n + 7 = O(n^2), O(n^6)$
- $f(n) = 9n + 10 = O(n)$
- $f(n) = f(n-1) + 1, f(0) = 0 = O(n)$

Complexități – notația Ω (big omega)

- Reprezintă o limită inferioară



Complexități – notația Ω

▣ Definiție:

$$f(n) \in \Omega(g(n)) \Leftrightarrow \exists c \in \mathbb{R}^+, n_0 \in \mathbb{R} \text{ s. t. :}$$

$$|f(n)| \geq c g(n) \quad \forall n \geq n_0$$

$$\lim_{n \rightarrow \infty} \sup \left| \frac{f(n)}{g(n)} \right| > 0$$

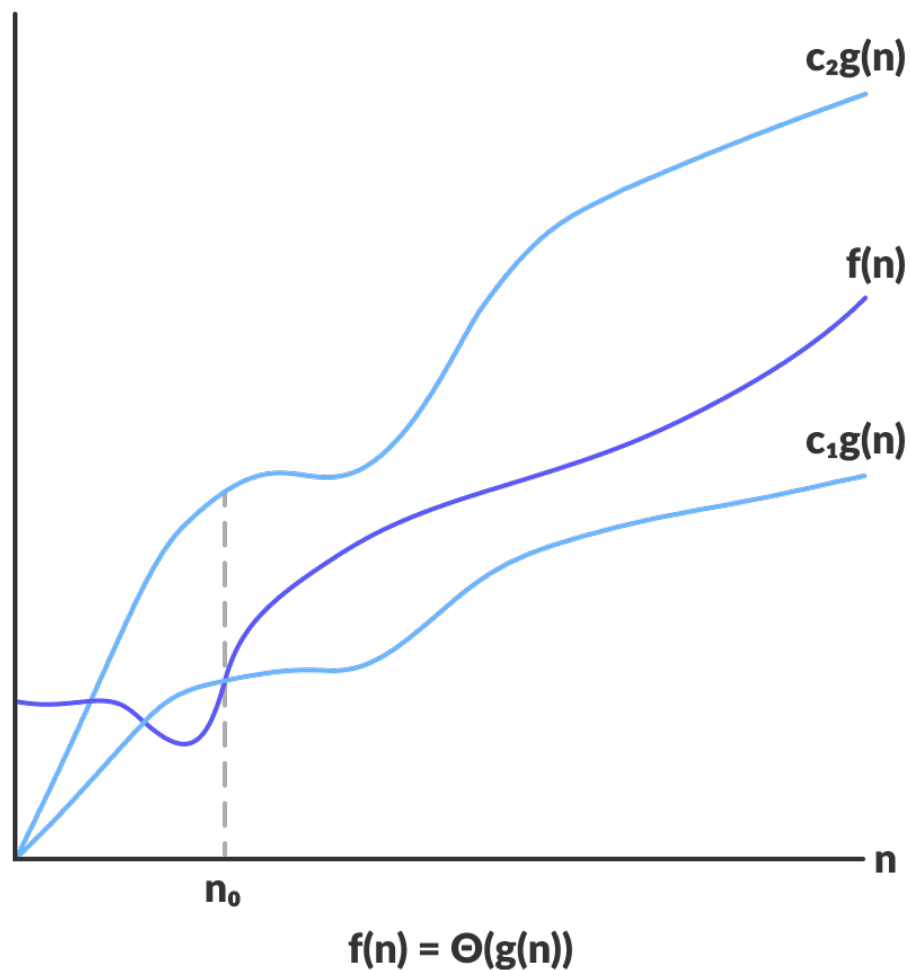
Complexități – notația Ω

□ Example:

- $f(n) = n^2 + 2n + 7 = \Omega(n^2), \Omega(n)$
- $f(n) = 9n + 10 = \Omega(n)$
- $f(n) = f(n-1) + 1, f(0) = 0 = \Omega(n)$

Complexități – notația Θ (big theta)

- Reprezintă o limită strânsă: și inferioară și superioară



Complexități – notația Θ

□ Definiție:

$$f(n) \in \Theta(g(n)) \iff \exists c_1, c_2 \in \mathbb{R}^+, n_0 \in \mathbb{R} \text{ s. t. :}$$

$$c_1 g(n) \leq f(n) \leq c_2 g(n) \quad \forall n \geq n_0$$

$$f(n) \in \Omega(g(n)), f(n) \in O(g(n))$$

Complexități – notația Θ

□ Example:

- $f(n) = n^2 + 2n + 7 = \Theta(n^2)$
- $f(n) = 9n + 10 = \Theta(n)$
- $f(n) = f(n-1) + 1, f(0) = 0 = \Theta(n)$

Sortări – de complexitate pătratică

- ❑ Bubblesort
- ❑ Insertion sort
- ❑ Selection sort

Sortări – de complexitate liniar-logaritmică

□ Quicksort

- Multe variante
- Discuție: PseudoQuicksort

□ Merge sort

□ Heapsort

□ Timsort

- Alege între insertion sort și merge sort
- Algoritmul folosit de Python și Java

□ Introsort

- Alege între quicksort, heapsort și insertion sort
- Algoritmul folosit de GNU C++ și .NET

Quickselect

- ❑ Care este al k-lea cel mai mic element dintr-un șir?
- ❑ Dorim să putem răspunde fără a sorta șirul.
- ❑ Quickselect:
 - În quicksort: <quicksort> pivot <quicksort>
 - Pivotul este pe poziția finală
 - Dacă poziția pivotului == k: pivotul e răspunsul
 - Dacă poziția pivotului < k: răspunsul e în dreapta
 - Altfel răspunsul e în stânga
 - Complexitate $O(n)$

Sortări în Python

- ❑ Sortări generice
- ❑ Key vs cmp
- ❑ Măsurarea timpului de execuție

Questions and Answers

Q & A