

SD Curs 5

26 Mar 2025

1. Limite inferioare pt. sortare
2. Metode de sortare în timp liniar
 - Count Sort
 - Radix Sort

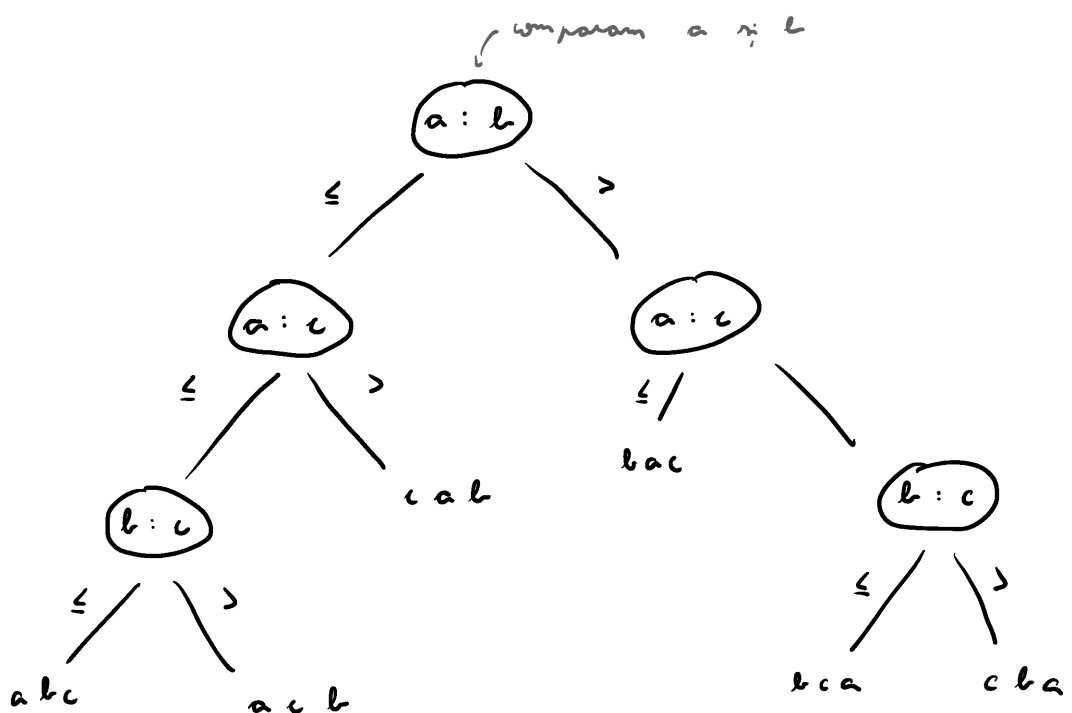
Limite inferioare pt. sortare

Teoremă Orice algoritm de sortare bazat pe comparații între chei folosește $\Omega(n \log n)$ comparații

Întrebare Cum micim orice algoritm de sortare bazat pe comparații între chei

Răspuns: Ca un arbore de decizii

Exemplu: Vrem să sortăm a, b, c



Arborele de decizie are $n!$ frunze

Timpul de rulare al algoritmului, în cazul cel mai defavorabil este **adâncimea arborelui**

Adâncimea unui arbore binar cu $n!$ frunze este **$\Omega(\log_2 n!) = \Omega(n \log n)$**

Metode de sortare în timp liniar

Problemă: Se dau n numere în intervalul

$$0 \dots n^2 - 1$$

Să se sorteze în $O(n)$

Count Sort $O(n + K)$

Fie n numere care pot lua K valori distincte

Exemplu:

0 1 1 0 2 1 2 0 1 1 2

$$n = 11$$

$$K = 3$$

0	1	2
3	5	3

0 0 0 1 1 1 1 2 2 2

Radix Sort

$$O(d(n+b))$$

n numere

d digits

b baza

// sortăm după unități \rightsquigarrow zeci \rightsquigarrow mii

307 \rightarrow bucket 7	180	100	100
239 \rightarrow b. 9	100	104	104
462 \rightarrow b. 2	462	306	180
306	222	307	222
549 \rightsquigarrow	793 \rightsquigarrow	222 \rightsquigarrow	239
104	104	239	306
180	306	549	307
666	666	462	462
793	307	666	549
222 \rightarrow b. 2	139	180	666
999 \rightarrow b. 9	549	793	793
100	999	999	999

Problema: Se dau n numere în intervalul
 $0 \dots n^2 - 1$

Să se sorteze în $O(n)$

Sol:

Idea 1: Quick Sort $\rightarrow O(n \log n)$

Idea 2: Radix Sort $\rightarrow O(\log n (n + 10))$
 $= O(n \log n)$

Idea 3:

n numere

b baza $\rightarrow O(d(n + b))$

d cifre

Transform numerele în baza n

$$d = 3$$

$$n = n$$

$$b = n$$

Fiecare număr x din intervalul $[0, n^2 - 1]$

se scrie ca $a_1 \cdot n^2 + a_2 \cdot n + a_3$

$$\Rightarrow O(3(n + n)) = O(n)$$