

# Probleme de Sortare

Andrei-Calin Georgescu

17.11.2018

Universitatea Politehnica Bucuresti  
Facultatea de Automatica si Calculatoare  
324CA

**Abstract.** Sortarea este unul dintre aspectele fundamentale ale stiintei calculatoarelor. Prezenta lucrare isi propune sa compare performantele a diferiti algoritmi de sortare(QuickSort, MergeSort, HeapSort), ca timp de executie si memorie utilizata pentru diferite segmente de date. Prin compararea eficientei acestor algoritmi se va determina cazul optim de folosire pentru fiecare in parte.

**Cuvinte Cheie:** QuickSort, MergeSort, HeapSort, eficienta de timp, eficienta de spatiu

## 1 Descrierea Problemei Rezolvate

### 1.1 Detalierea Problemei

Sortarea sta la baza mai multor algoritmi complecsi din programarea calculatoarelor(Divide et Impera, Binary Search etc.). Astfel este importanta gasirea unui algoritm de sortare care sa satisfaca cerintele impuse de situatia cu care ne confruntam.

Asadar, lucrarea va detalia sortarea mai multor segmente de date (alese pentru a testa cazurile favorabile/medii/defavorabile pentru fiecare algoritm) si va analiza algoritmii alesi din punct de vedere al timpului de executie si a memoriei necesare.

### 1.2 Aplicatii practice

Prin aceasta lucrare se doreste crearea unui ghid pentru aplicarea algoritmilor de sortare potriviti problemei care necesita solutionare, in speta in functie de setul de date primit si limitarile legate de memorie.

## 2 Specificarea solutiilor alese

Solutiile alese pornesc de la algoritmi care pot fi folositi in scop general si avanseaza catre algoritmi utili intr-o anumita situatie de utilizare.

- QuickSort este general folosit, insa pentru anumite seturi de date timpul de executie devine indelungat
- MergeSort este folosit cand datele de sortat se afla in diferite locatii de memorie(cache, memorie principala, memorie externa). Timp de rulare mai scurt ca QuickSort
- HeapSort este usor de implementat si devine alegerea naturala pentru secvente mici si mijlocii de date

## 3 Criterii de evaluare pentru solutia propusa

Pentru determinarea eficientei, se vor rula cei trei algoritmi pe cate 4 seturi de date fiecare. Seturile de date vor consta in vectori de date care vor indeplini alternativ urmatoarele conditii: date complet aleatorii, date partial sortate, date sortate complet si date sortate complet, insa in ordine descrescatoare.

Vom determina timpul de executie pentru fiecare set de date in parte si il vom inregistra intr-un tabel comparativ

Astfel se va testa cazul cel mai bun, cel mediu si cel defavorabil de utilizare pentru fiecare algoritm in parte.

Se va determina complexitatea fiecaruia si se va intocmi o analiza amortizata pentru fiecare in parte. In acest mod se pot testa algoritmii in diferite situatii de utilizare si se poate intocmi o analiza echilibrata de utilizare pentru cei trei algoritmi alesi.

## Referinte

- [1] Edosomwan Joseph. *Sorting Algorithm: Analysis and Comparison Performance*, Lambert Publishing, 19 Jul 2012
- [2] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. *Introduction to Algorithms, 3rd Edition*, The MIT Press