

O COMPARAȚIE TEORETICĂ ȘI EXPERIMENTALĂ A UNOR METODE DE SORTARE

Rancov Larisa
Facultatea de Matematică și Informatică
Universitatea de Vest Timișoara, România
larisa.rancov03@e-uvt.ro

27 Mai 2023

Introducere

Ce este sortarea?

Care este rolul algoritmilor de sortare?

Motivația acestei lucrări?

Fundamentarea Teoretică

Algoritmii pe care am ales să îi analizez:

- Bubblesort
- Inserion Sort
- Selection Sort
- Quicksort
- Merge Sort
- Radix Sort
- Heap Sort

Configurarea și desfășurarea experimentelor

- Limbajul de programare folosit: C.
- Tipurile de seturi de date: sortate, nesortate, parțial sortate.
- Generarea setului de date: funcția 'random' din Python.
- Criteriul de evaluare: timpul de rulare al algoritmilor.

Rezultate și Analiză

Table: Compararea timpilor de rulare ai algoritmilor de sortare

Algoritmul	Caz favorabil	Caz mediu	Caz nefavorabil
Insertia	0.07	1.906	2.455
Selecția	2.779	2.803	2.861
Heap	0.003	0.007	0.016
BubbleSort	2.644	6.34	7.632
Merge	0.008	0.016	0.016
Quick	Un număr ciudat	0.006	0.015
Radix	0.014	0.005	0.019

Table: Compararea algoritmilor de sortare

	Număr de elemente			
	100	1000	10000	50000
Insertion Sort	0,002	0,006	0,008	2,455
Selection Sort	0,002	0,006	0,007	2,861
Merge Sort	0,0001	0,0002	0,004	0,016
QuickSort	0,0001	0,0001	0,003	0,015
Heap Sort	0,0001	0,0003	0,001	0,016
Bubblesort	0,001	0,011	0,037	7,632
Radix Sort	0,0001	0,0001	0,006	0,019

Discuție și Concluzii

Cel mai eficient algoritm de sortare: QuickSort.

Criteriile alegerii unui algoritm de sortare:

- caracteristicile datelor care urmează a fi sortate
- complexitatea algoritmilor
- metoda de implementare