

RA03 - Trabalho Ordenação

ALUNO: Fernando Moro

Relatório -----

O algoritmo de ordenação Quick Sort superou o Bubble Sort e o Shell Sort nos quesitos avaliados(Tempo de execução, Num. de Trocas e Num. de iterações).

Já o Bubble Sort teve tempos de execução mais longos e um número de trocas maior, enquanto o Shell Sort, apesar de poucas trocas, teve tempos de execução mais longos, principalmente com arrays maiores.

Portanto o Quick Sort parece ser o melhor algoritmo de ordenação, com um bom tempo de execução e pouco numero de trocas, enquanto o Bubble Sort aparenta ser o pior dentre os analisados.

Resultados obtidos -----

Bubble Sort :

--> Array com 50 elementos:

Tempo de execução: 0 ms

Número de trocas: 512

Número de iterações: 1225

--> Array com 500 elementos:

Tempo de execução: 0 ms

Número de trocas: 63252

Número de iterações: 124750

--> Array com 1000 elementos:

Tempo de execução: 16 ms

Número de trocas: 251695

Número de iterações: 499500

--> Array com 5000 elementos:

Tempo de execução: 23 ms

Número de trocas: 6171193

Número de iterações: 12497500

--> Array com 10000 elementos:

Tempo de execução: 91 ms

Número de trocas: 25126680

Número de iterações: 49995000

Quick Sort:

--> Array com 50 elementos:

Tempo de execução: 0 ms

Número de trocas: 1250

Número de iterações: 1205

--> Array com 500 elementos:

Tempo de execução: 3 ms

Número de trocas: 96582

Número de iterações: 96311

--> Array com 1000 elementos:

Tempo de execução: 2 ms

Número de trocas: 317228

Número de iterações: 317028

--> Array com 5000 elementos:

Tempo de execução: 4 ms

Número de trocas: 2487550

Número de iterações: 2496010

--> Array com 10000 elementos:

Tempo de execução: 19 ms

Número de trocas: 4982124

Número de iterações: 5023059

Shell Sort:

--> Array com 50 elementos:

Tempo de execução: 0 ms

Número de trocas: 0

Número de iterações: 203

--> Array com 500 elementos:

Tempo de execução: 0 ms

Número de trocas: 0

Número de iterações: 3506

--> Array com 1000 elementos:

Tempo de execução: 0 ms

Número de trocas: 0

Número de iterações: 8006

--> Array com 5000 elementos:

Tempo de execução: 1 ms

Número de trocas: 0

Número de iterações: 55005

--> Array com 10000 elementos:

Tempo de execução: 0 ms

Número de trocas: 0

Número de iterações: 120005