

Relatório RA04

Gustavo Albiero, Gustavo Fraiz

November 4, 2023

1 Introdução

Neste trabalho foram implementados e comparados diferentes algoritmos de ordenação de dados com conjuntos de diferentes tamanhos para analisar o desempenho dos algoritmos:

Quick Sort
Bubble Sort
Merge Sort
Selection Sort

2 Ordenações

Para comparar as estratégias de ordenação, será analisado o número de trocas, iterações e uma média do tempo de 5 diferentes execuções do programa para cada ordenação, todos comparados com os mesmos números (gerados pela mesma seed), porém com vetores de 50, 500, 1000, 5000 e 10000 elementos.

Ordenação de 50 Elementos	Trocas	Iterações	Tempo de Execução(ms)
QuickSort	184	317	7,2
BubbleSort	602	2450	7,6
MergeSort	286	433	5,6
SelectionSort	49	1254	6,2

Ordenação de 500 Elementos	Trocas	Iterações	Tempo de Execução(ms)
QuickSort	2571	5330	15
BubbleSort	64601	249500	20,8
MergeSort	4488	5985	14,4
SelectionSort	499	125249	18,6

Ordenação de 1000 Elementos	Trocas	Iterações	Tempo de Execução(ms)
QuickSort	5934	12172	19,6
BubbleSort	254762	999000	29,2
MergeSort	9976	12973	21,2
SelectionSort	999	500499	23

Ordenação de 5000 Elementos	Trocas	Iterações	Tempo de Execução(ms)
QuickSort	40994	77583	43,8
BubbleSort	6280172	24995000	102,2
MergeSort	61808	76805	46,4
SelectionSort	4999	12502499	71,6

Ordenação de 10000 Elementos	Trocas	Iterações	Tempo de Execução(ms)
QuickSort	82227	170171	66,4
BubbleSort	24891921	99990000	309
MergeSort	133616	163613	74,8
SelectionSort	9999	50004999	133,2

*A máquina usada para as análises possui um processador Intel i7 7700 e 16 GB de Memória RAM.

3 Conclusões Finais

Após realizar as análises foi possível observar o desempenho particular de cada estratégia de ordenação de uma maneira geral. A primeira coisa que pode ser observada se trata do BubbleSort em comparação aos outros algoritmos, sendo em todas as ocasiões o método mais lento de se fazer uma ordenação, que fica evidente quando comparado aos outros métodos no caso de ordenação de 10000 elementos, pois o BubbleSort possui uma média de tempo de execução mais do que o dobro maior que a SelectionSort e cerca de 4,6 vezes maior do que o QuickSort.

Agora falando sobre características dos outros algoritmos, a partir das análises, observamos que o número de trocas do SelectionSort sempre foi o mais baixo ao longo de todos os testes, porém possui uma desvantagem no desempenho quando se trata do número de iterações.

A Estratégia MergeSort apresenta um número de trocas e iterações bastante similar durante todo o experimento e também é o algoritmo mais rápido em

comparação aos outros durante as análises com menos elementos.

O Algoritmo que se destacou no experimento foi o QuickSort, que quando se trata de velocidade, venceu em quase todos os cenários por ser uma estratégia muito bem planejada e que possui o menor número de trocas e iterações no total de cada teste e sem aumentar muito o tempo de execução apesar do número de elementos aumentar em 500 vezes ao longo do experimento.