Algoritmos de ordenação

Luize Cunha Duarte
luize.duarte@ufpr.br
Universidade Federal do Paraná
2023

1. Introdução.

O trabalho tem o objetivo de comparar o desempenho dos algoritmos Busca Sequencial, Busca Binária, Insertion Sort, Selection Sort, Merge Sort, Quick Sort e Heap Sort. Os parâmetros de comparação são a quantidade de comparações feitas entre elementos do vetor e o tempo de execução em segundos (CPU time). Os testes para os algoritmos de ordenação foram divididos em: vetor já ordenado, vetor ordenado ao contrário e vetor aleatorizado. Já os de busca foram analisados considerando os piores casos.

2. Algoritmos de busca.

1.1. Busca Sequencial.

O vetor é percorrido do final em direção ao começo até encontrar o valor desejado. Portanto, o pior caso ocorre quando o valor que deseja está na primeira casa, sendo o número de comparações o equivalente ao tamanho do vetor.

O tempo em segundos resultante do teste para um vetor de tamanho 50 000 foi de 0,003826.

1.2. Busca Binária.

De maneira mais eficiente, o algoritmo consegue realizar as operações em 0,000001 segundos, levando 32 comparações quando o valor desejado se encontra na primeira casa do vetor.

3. Algoritmos de ordenação.

Para os cálculos, foram utilizados vetores organizados de três formas distintas, todos de tamanho 50 000.

2.1. Insertion Sort.

	comparações	tempo de CPU
ordenado	49999	0,000992
inverso	1250024999	6,338677
aleatorizado	629267782	3,211485

Seu pior caso será, portanto, um vetor ordenado em ordem inversa.

2.2. Selection Sort.

	comparações	tempo de CPU
ordenado	1249975000	2,44025
inverso	1249975000	2,342727
aleatorizado	1249975000	2,444689

O algoritmo faz sempre o mesmo número de comparações, uma vez que percorre o vetor diversas vezes procurando o menor elemento.

2.3. Merge Sort.

	comparações	tempo de CPU
ordenado	784464	0,004678
inverso	784464	0,004478
aleatorizado	784464	0,007746

O algoritmo faz sempre o mesmo número de comparações, devido a criação de seu vetor auxiliar ordenado.

2.4. Quick Sort.

	comparações	tempo de CPU
ordenado	1250024999	6,874612
inverso	1250024999	4,672888
aleatorizado	981031	0,007089

Segundo a tabela, seu pior caso é quando os elementos estão ordenados de alguma forma. Mas, também é possível descrever seu pior caso como quando o tamanho das partições realizadas equivalem a 0 e o tamanho do vetor - 1.

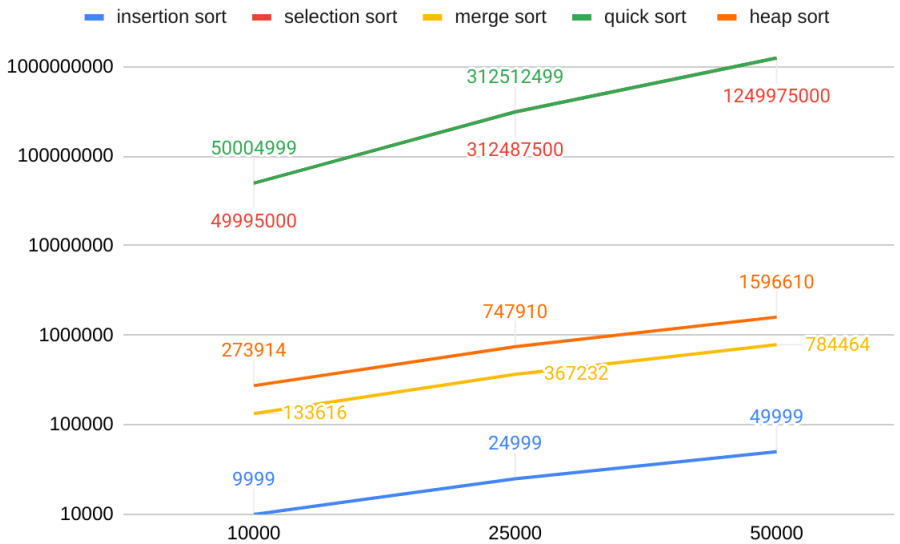
2.5 Heap Sort.

	comparações	tempo de CPU
ordenado	1596610	0,009815
inverso	1447786	0,009409
aleatorizado	1524508	0,012330

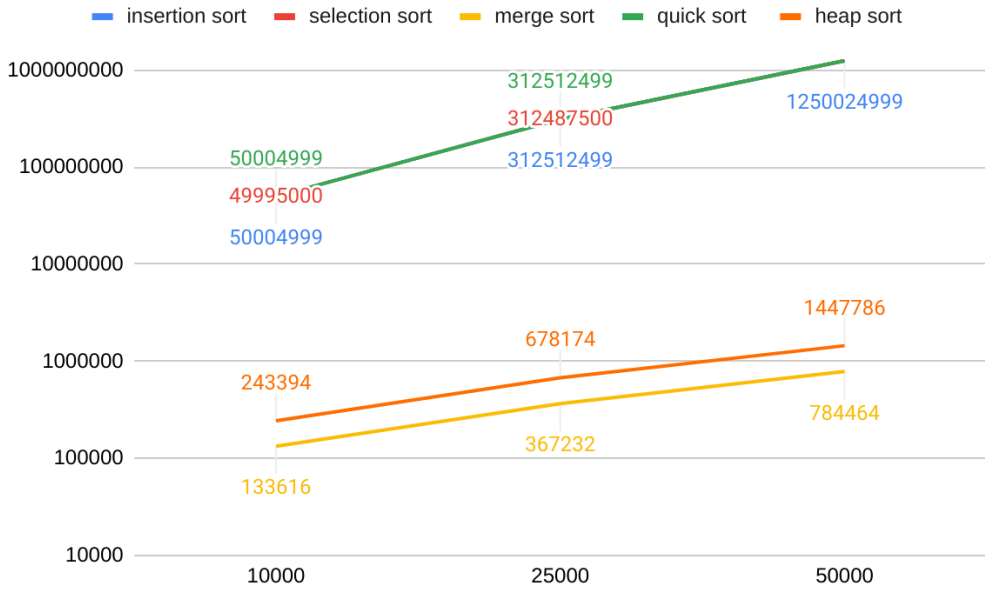
O algoritmo faz aproximadamente o mesmo número de comparações.

4. Gráficos comparativos.

4.1. Número de comparações realizadas para vetores já ordenados entre os algoritmos.



4.2. Número de comparações realizadas para vetores ordenados ao contrário entre os algoritmos.



4.3. Número de comparações realizadas para vetores aleatorizados entre os algoritmos.