

Thiago I. Lima

GRR:20204397

InsertionSort, SelectionSort, MergeSort, QuickSort, HeapSort, foram os algoritmos de ordenação de vetores que verificamos seu funcionamento durante a aula, juntamente com algoritmos de Busca, que são BuscaSequencial e BuscaBinária. Neste relatório vamos observar as diferenças entre os algoritmos de ordenação e as diferenças entre os algoritmos de busca usando Windows 10 como Sistema Operacional.

Vamos usar um vetor de  $n$  posições preenchido aleatoriamente com a função *rand()*, com número entre 0 e 100, em seguida ordenaremos usando todos os métodos de ordenação, fazendo a medição de tempo com a função *clock()* e também quantidade de comparações que cada algoritmo fez para realizar a ordenação. Para as pesquisas usaremos os vetores já ordenados.

**Teste 1:** No primeiro teste,  $n$  será igual a 1000.

- InsertionSort teve um tempo de 0.001000 segundos realizando 1000 comparações;
- SelectionSort teve um tempo de 0.001000 segundos realizando 499500 comparações;
- MergeSort teve um tempo de 0.000000 segundos realizando 9976 comparações;
- QuickSort teve um tempo de 0.000000 segundos realizando 12736 comparações.
- HeapSort teve um tempo de 0.003000 segundos realizando 19144 comparações;

**Teste 2:** No segundo teste,  $n$  será igual a 15000.

- InsertionSort teve um tempo de 0.103000 segundos realizando 15000 comparações;
- SelectionSort teve um tempo de 0.225000 segundos realizando 112492500 comparações;
- MergeSort teve um tempo de 0.000000 segundos realizando 208616 comparações;
- QuickSort teve um tempo de 0.016000 segundos realizando 1272509 comparações.

- HeapSort teve um tempo de 0.000000 segundos realizando 401630 comparações;

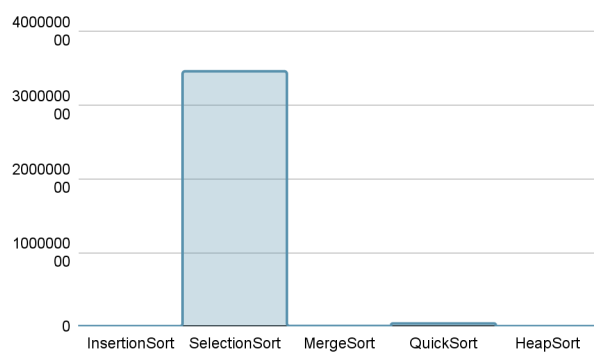
**Teste 3:** No terceiro teste, **n** será igual a 43000.

- InsertionSort teve um tempo de 0.919000 segundos realizando 43000 comparações;
- SelectionSort teve um tempo de 1.746000 segundos realizando 924478500 comparações;
- MergeSort teve um tempo de 0.005000 segundos realizando 665464 comparações;
- QuickSort teve um tempo de 0.053000 segundos realizando 9662909 comparações;
- HeapSort teve um tempo de 0.001000 segundos realizando 1283798 comparações;

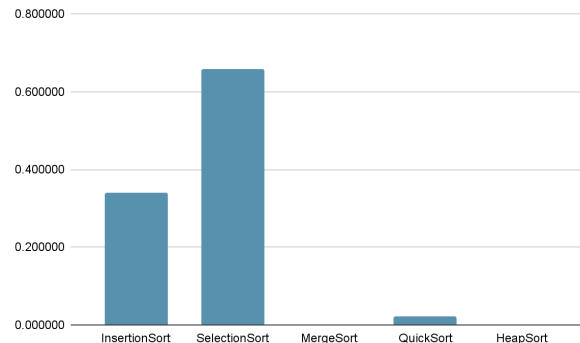
Com as medições realizadas em três testes diferentes, onde aumentamos gradativamente a dificuldade para o algoritmo, vamos a seguir calcular a média aritmética de tempo e de comparações de cada método e representá-las graficamente.

### Gráficos ordenação:

**Comparações por Algoritmo:**



**Tempo de execução por Algoritmo:**



Para os algoritmos de busca, realizaremos novamente 2 testes, agora nos vetores já ordenados, onde pesquisamos um valor pré-determinado em cada teste.

### Teste 1:

Valor pesquisado: 97

#### Sequencial:

Tempo total: 0.000000

Comparações: 878

Indice no Vetor: 42122

#### Binario:

Tempo total: 0.000000

Comparações: 13

Indice no Vetor: 41993

### Teste 2:

Valor pesquisado: 1

#### Sequencial:

Tempo total: 0.000000

Comparações: 12422

Indice no Vetor: 892

#### Binario:

Tempo total: 0.000000

Comparações: 11

Indice no Vetor: 892

Media de Comparações

