Relatório de Desempenho em relação a tempo, número de trocas e número de comparações

Métodos de ordenação:

1. O Bubble Sort, também conhecido como ordenação por flutuação, é um algoritmo de ordenação bastante simples. Sua abordagem consiste em percorrer repetidamente um conjunto de elementos e, a cada iteração, movimentar o maior elemento da sequência para a posição mais alta.
2. O Insertion Sort é um algoritmo de ordenação que constrói uma matriz final adicionando um elemento de cada vez, realizando uma inserção por vez. Embora seja classificado como um algoritmo de ordenação quadrática, é altamente eficaz para problemas com entradas pequenas, destacando-se como o mais eficiente entre os algoritmos dessa categoria.
3. O Selection Sort funciona selecionando o menor valor presente no vetor e movendo-o para a primeira posição. Em seguida, seleciona o próximo menor valor e o posiciona na segunda posição, e assim por diante, até que todo o vetor esteja ordenado.

O resultado dos testes para arrays de elementos de 10, 100 e 1000 elementos ficou da seguinte forma:

Array de 10 elementos:

Bubble Sort:

Tempo de execução: 3363 milissegundos.

Número de trocas: 24

Número de comparações: 45

Insertion Sort:

Tempo de execução: 1614 milissegundos.

Número de trocas: 9

Número de comparações: 18

Selection Sort:

Tempo de execução: 3164 milissegundos.

Número de trocas: 9

Número de comparações: 45

Array de 100 elementos:

Bubble Sort:

Tempo de execução: 301143 milissegundos.

Número de trocas: 1732

Número de comparações: 4950

Insertion Sort:

Tempo de execução: 5445 milissegundos.

Número de trocas: 99

Número de comparações: 198

Selection Sort:

Tempo de execução: 196377 milissegundos.

Número de trocas: 99

Número de comparações: 4950

Array de 1000 elementos:

Bubble Sort:

Tempo de execução: 61866819 milissegundos.

Número de trocas: 47041

Número de comparações: 499500

Insertion Sort:

Tempo de execução: 60804 milissegundos.

Número de trocas: 999

Número de comparações: 1998

Selection Sort:

Tempo de execução: 5644291 milissegundos.

Número de trocas: 999

Número de comparações: 599500

Conclusões:

Tempo de Execução: Para arrays de todos os tamanhos, o Insertion Sort demonstrou ser o mais rápido, seguido pelo Selection Sort e por último Bubble Sort.

Número de Trocas: O Bubble Sort teve o maior número de trocas para todos os tamanhos de arrays, o Insertion e o Selection tiveram o mesmo número de trocas em todos os casos.

Número de comparações: Os algoritmos Bubble Sort e Selection Sort tiveram o mesmo número de comparações em todos os casos, sendo superior ao algoritmo Insertion Sort.

Com essas informações pode se afirmar que o algoritmo Insertion Sort tem o melhor desempenho entre os três para os casos de arrays com 10, 100 e 1000 elementos, tendo o menor tempo de execução, o menor número de trocas e o menor número de comparações em todos os casos.