

Tecnologia em Análise e Desenvolvimento de Sistemas.

Aluno: Luís Lucilandio de Queiroz Mesquita.

Professor: Leandro Luttiane.

Matéria: Algoritmos.

Comparação de Tempo dos Algoritmos de Ordenação.

- Bubble Sort.

O algoritmo de “Ordenação por Bolha”, ou “Bubble Sort”, recebeu este nome pela forma em que ele realiza a troca de valores: “Como se estivesse borbulhando”, para realizar a sua função. Os elementos maiores são mais leves, e sobem como bolhas até suas posições corretas. Porém, se tratando de eficiência, este algoritmo não é muito utilizado atualmente. A complexidade desse algoritmo é de ordem quadrática. $O(n)^2$

Em seu melhor caso, o algoritmo executa n operações relevantes, onde n representa o número de elementos no vetor. No pior caso, são feitas n^2 operações.

ANALISANDO OS CASOS DE TEMPO.

A média de 100 vetores com 100 elementos pode chegar a **120.080 ns** (0,12008 em milissegundos). A média dos vetores de 10.000 elementos foi de **706 ms**; A média do caso final de 50.000 elementos ele chegou à **16.144 ms**, e assim sucessivamente. (Para conferir todos os casos, faça uma análise do gráfico, ou dos arquivos).

Em outras palavras, a tendência deste algoritmo é crescer no tempo conforme a quantidade de elementos. Por este motivo, ele não é recomendado para programas que precisem de velocidade e operem com quantidade elevada de dados.

- Selection Sort.

A ideia do algoritmo consiste em ordenar os vetores “selecionando” a cada iteração os menores itens possíveis e os colocam da esquerda para a direita. Ao contrário do Bubble Sort, ele ordena o vetor pelo começo. O caso da seleção do menor valor será mais eficiente em vetores pequenos, em vetores de grandes proporções, o algoritmo levará mais tempo para selecionar e começar a ordenação.

Para análise assintótica, podemos ignorar os valores constantes. O que importa mesmo $n^2 / 2$, pois conforme a lista crescer, este valor irá aumentar quadraticamente. Por isso, o algoritmo selection sort possui tempo de execução $O(n^2)$.

ANALISANDO OS CASOS DE TEMPO.

A média dos 100 vetores com 100 elementos chegou a **40.620 ns**. A média dos vetores de 10.000 elementos chegou a **243 ms**. A média do caso final de 50.000 elementos chegou a **6013 ms**. (Para conferir todos os casos, faça uma análise do gráfico, ou dos arquivos).

- Insertion Sort.

É o algoritmo de ordenação que, dado uma estrutura (array, lista) constrói uma matriz final com um elemento de cada vez, uma inserção por vez. Assim como algoritmos de ordenação quadrática, é bastante eficiente para problemas com pequenas entradas, sendo o mais eficiente entre os algoritmos desta ordem de classificação. Este algoritmo também possui complexidade $O(n)^2$.

Mesmo possuindo esta complexidade, o Insertion Sort é o melhor quando se trata de tempo de execução, em comparação com Bubble Sort e Selection Sort.

Melhor caso: quando os elementos já estão ordenados: $O(n)$;

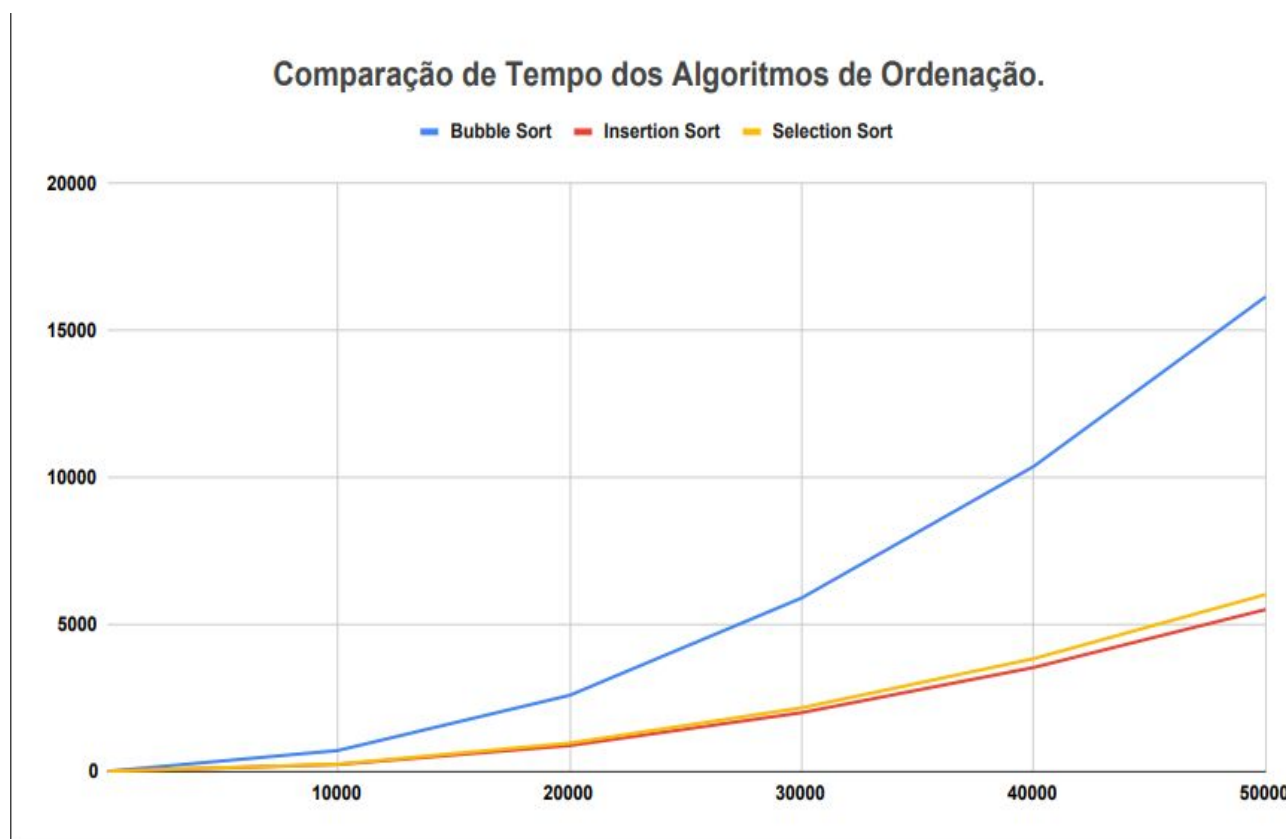
Caso médio: quando parte dos elementos já está ordenada: $O(n^2/4)$;

Pior caso: quando os elementos estão ordenados em ordem inversa: $O(n)^2$.

ANALISANDO OS CASOS DE TEMPO.

A média dos 100 vetores com 100 elementos chegou a **11.995 ns**. A média dos vetores de 10.000 elementos chegou a **230 ms**. A média do caso final de 50.000 elementos chegou a **5503 ms**. (Para conferir todos os casos, faça uma análise do gráfico, ou dos arquivos).

GRÁFICO DE COMPARAÇÃO DE TEMPO DOS ALGORITMOS..



Fonte: Autoria própria.

CONCLUSÃO:

Foi notório que o algoritmo de ordenação Bubble Sort possui tendência a crescer conforme seus dados e números de elementos. Apesar de ser “Eficiente” em vetores de pequenas proporções, no caso de um sistema gigantesco ele não seria o melhor para utilizar.

O nível dos demais algoritmos mostram que possuem uma ordenação crescente, ou seja, tendem a aumentar seu nível de complexidade conforme seu número de elementos. Porém, mais eficientes que o Bubble Sort.

O resultado desses dados significa o quão importante é desenvolver aplicações otimizadas para que tenham boas funcionalidades, independentemente da máquina do usuário.

REFERÊNCIAS:

https://pt.wikipedia.org/wiki/Bubble_sort

https://medium.com/@henriquebraga_18075/algoritmos-de-ordena%C3%A7%C3%A3o-ii-selection-sort-8ee4234deb10

https://pt.wikipedia.org/wiki/Selection_sort

<http://www.devfuria.com.br/logica-de-programacao/introducao-ao-algoritmo-de-ordenacao-insertion-sort/>

https://pt.wikipedia.org/wiki/Insertion_sort