



ESTRUTURA DE DADOS

Aula 10 – Ordenação I

Prof. Rodrigo Maciel

ORDENAÇÃO

- Após uma base de dados estar construída, pode ser necessário ordená-la, como:
 - Nomes em ordem alfabética
 - Alunos por nota
 - Clientes por CEP
 - Vendas por preço
 - Cidades em ordem crescente de população
- Ordenar dados pode ser um passo preliminar para pesquisá-los;
- Existem várias razões para se ordenar uma sequência, uma delas é a possibilidade de acessar seus dados de modo mais eficiente.

BUBBLE SORT - BOLHA

- Notavelmente lento e é o mais simples dos algoritmos de ordenação;
- Funcionamento:
 - Comparação de dois números
 - Se o da esquerda for maior, os elementos devem ser trocados
 - Desloca-se uma posição à direita
- À medida que o algoritmo avança, os itens maiores “surgem como uma bolha” na extremidade superior do vetor;
- Visualização on-line: <https://visualgo.net/en/sorting>

BUBBLE SORT - BOLHA

- O algoritmo com 10 elementos faz 9 comparações na primeira passagem, 8 na segunda, 7 na terceira, etc ($n - 1$, $n - 2$, $n - 3$);
- Para 10 itens: $9 + 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1 = 45$ comparações;
- O algoritmo faz em média $N^2/2$ comparações;
- Há menos trocas do que há comparações, pois dois elementos serão trocados somente se precisarem;
- Se os dados forem aleatórios, uma troca será necessária mais ou menos $N^2/4$;
- No pior caso, com os dados em modo inverso, uma troca será necessária a cada comparação.

BUBBLE SORT - ALGORITMO

buble_sort(array):

 n <- tamanho(array)

para cada *i* no range(n) **faça:**

para cada *j* no range(0, n - *i* - 1) **faça:**

se array[*j*] > array[*j* + 1] **então:**

 temp <- array[*j*]

 array[*j*] <- array[*j* + 1]

 array[*j* + 1] <- temp

return array

INSERTION SORT

- É cerca de duas vezes mais rápido que a ordenação pelo método da bolha em arrays randômicos;
- Pode ser mais lento que o método da bolha caso o array já esteja ordenado de forma inversa;
- Funcionamento:
 - Há um marcador em algum lugar no meio do vetor
 - Os elementos à esquerda do marcador estão parcialmente ordenados (estão ordenados entre eles, porém não estão em suas posições finais)
 - Os elementos à direita do marcador não estão ordenados
- Visualização on-line: <https://visualgo.net/en/sorting>

INSERTION SORT

- Na primeira passagem, é comparado no máximo um item. Na segunda passagem, máximo de dois itens, etc.
 - $1 + 2 + 3 + \dots + N - 1 = N*(N-1)/2$
- Como em cada passagem uma média de apenas metade do número máximo de itens é de fato comparada antes do ponto de inserção ser encontrado, então:
 - $N*(N-1)/4$
- O número de cópias é aproximadamente o mesmo que o número de comparações;
- Para dados em ordem inversa, todas as comparações e deslocamentos são executados, sendo mais lento que o método bolha.

INSERTION SORT - ALGORITMO

insertion_sort(array):

$n \leftarrow \text{tamanho}(\text{array})$

para cada i **no** range(1, n) **faça:**

$\text{marcado} \leftarrow \text{array}[i]$

$j \leftarrow i - 1$

enquanto $j \geq 0$ **E** $\text{marcado} < \text{array}[j]$ **então:**

$\text{array}[j + 1] \leftarrow \text{array}[j]$

$j \leftarrow j - 1$

$\text{array}[j + 1] \leftarrow \text{marcado}$

return array