
Ordenação de Dados

Método Bubble Sort

Ordenação por Bolha

- *Vantagens:*
 - Algoritmo simples
 - Algoritmo estável
- *Desvantagens:*
 - O fato de o arquivo já estar ordenado não ajuda em nada, pois o custo continua quadrático.

Método Bolha

Os elementos vão “borbulhando” a cada iteração do método até a posição correta para ordenação da lista

O método poderia parar quando nenhum elemento borbulhace/trocasse de posição

Como os elementos são trocados (borbulhados) frequentemente, há um alto custo de troca de elementos

Ordenação de Dados

- *Processo bastante utilizado na computação de uma estrutura de dados*
- *Dados ordenados garantem uma melhor performance de pesquisa a uma ED*
 - busca sequencial
 - evita a varredura completa de uma lista de dados
 - busca binária
 - só é possível se os dados estão ordenados
 - apresenta baixa complexidade

Compromisso

- *“A complexidade da ordenação da ED não deve exceder a complexidade da computação a ser feita na ED sem o processo de ordenação”*
- *Exemplo: deseja-se realizar uma única pesquisa a um vetor*
 - *busca sequencial $\Rightarrow O(n)$*
 - *ordenação $\Rightarrow O(n \log n)$*
 - *Não vale a pena ordenar!*

Considerações

- *Dados estão mantidos em um vetor*
- *Elemento do vetor*
 - objeto que possui um atributo **chave** que deve ser mantido ordenado
- *Um método **troca(x,y)** realiza a troca dos elementos presentes nas posições **x** e **y** do vetor*
- *Para fins de exemplo, números inteiros serão utilizados como elementos*

Métodos de Ordenação

- **Ordenação por troca**
 - BubbleSort (método da bolha)
 - QuickSort (método da troca e partição)
- **Ordenação por inserção**
 - InsertionSort (método da inserção direta)
 - BinaryInsertionSort (método da inserção direta binária)
- **Ordenação por seleção**
 - SelectionSort (método da seleção direta)
 - HeapSort (método da seleção em árvore)
- **Outros métodos**
 - MergeSort (método da intercalação)
 - BucketSort (método da distribuição de chave)

Métodos de Ordenação Simples

- São três
 - *BubbleSort*
 - *InsertionSort*
 - *SelectionSort*
- Características
 - fácil implementação
 - alta complexidade
 - comparações ocorrem sempre entre posições adjacentes do vetor

“Revisão” de Somatória

- *Propriedade 1 (P1)*

$$\sum_{i=1}^n i = \frac{n(n+1)}{2}$$

- *Propriedade 2 (P2)*

$$\sum_{i=1}^n k i = k \sum_{i=1}^n i$$

BubbleSort

- *BubbleSort* é um *método simples de troca*
 - ordena através de sucessivas trocas entre pares de elementos do vetor
- *Características*
 - realiza varreduras no vetor, trocando *pares adjacentes* de elementos sempre que o próximo elemento for menor que o anterior
 - após uma varredura, o maior elemento está corretamente posicionado no vetor e não precisa mais ser comparado
 - após a *i-ésima* varredura, os *i* maiores elementos estão ordenados

BubbleSort

- *Simulação de funcionamento*

<http://math.hws.edu/TMCM/java/xSortLab>

BubbleSort - Complexidade

- Para um vetor de n elementos, $n - 1$ varreduras são feitas para acertar todos os elementos
 $n = 5$

4	9	2	5	1	3

BubbleSort - Ordenação

4	9	2	5	1	3
4	9	2	5	1	3

BubbleSort - Ordenação

4	9	2	5	1	3
4	9	2	5	1	3
2	9	4	5	1	3

BubbleSort - Ordenação

4	9	2	5	1	3
4	9	2	5	1	3
2	9	4	5	1	3
2	9	4	5	1	3

BubbleSort - Ordenação

4	9	2	5	1	3
4	9	2	5	1	3
2	9	4	5	1	3
2	9	4	5	1	3
1	9	4	5	2	3

BubbleSort - Ordenação

4	9	2	5	1	3
4	9	2	5	1	3
2	9	4	5	1	3
2	9	4	5	1	3
1	9	4	5	2	3
1	9	4	5	2	3

BubbleSort - Ordenação

4	9	2	5	1	3
4	9	2	5	1	3
2	9	4	5	1	3
2	9	4	5	1	3
1	9	4	5	2	3
1	9	4	5	2	3
1	4	9	5	2	3

BubbleSort - Ordenação

4	9	2	5	1	3
4	9	2	5	1	3
2	9	4	5	1	3
2	9	4	5	1	3
1	9	4	5	2	3
1	9	4	5	2	3
1	4	9	5	2	3
1	4	9	5	2	3

BubbleSort - Ordenação

4	9	2	5	1	3
4	9	2	5	1	3
2	9	4	5	1	3
2	9	4	5	1	3
1	9	4	5	2	3
1	9	4	5	2	3
1	4	9	5	2	3
1	4	9	5	2	3
1	2	9	5	4	3

BubbleSort - Ordenação

4	9	2	5	1	3
4	9	2	5	1	3
2	9	4	5	1	3
2	9	4	5	1	3
1	9	4	5	2	3
1	9	4	5	2	3
1	4	9	5	2	3
1	4	9	5	2	3
1	2	9	5	4	3
1	2	9	5	4	3

BubbleSort - Ordenação

4	9	2	5	1	3	es
4	9	2	5	1	3	es
2	9	4	5	1	3	es
2	9	4	5	1	3	o
1	9	4	5	2	3	
1	9	4	5	2	3	
1	4	9	5	2	3	
1	4	9	5	2	3	
1	2	9	5	4	3	
1	2	5	9	4	3	

BubbleSort - Ordenação

4	9	2	5	1	3
4	9	2	5	1	3
2	9	4	5	1	3
2	9	4	5	1	3
1	9	4	5	2	3
1	9	4	5	2	3
1	4	9	5	2	3
1	4	9	5	2	3
1	2	9	5	4	3
1	2	4	9	5	3

BubbleSort - Ordenação

1	9	4	5	2	3
1	4	9	5	2	3
1	4	9	5	2	3
1	2	9	5	4	3
1	2	4	9	5	3
1	2	3	9	5	4

BubbleSort - Ordenação

1	9	4	5	2	3
1	4	9	5	2	3
1	4	9	5	2	3
1	2	9	5	4	3
1	2	4	9	5	3
1	2	3	9	5	4
1	2	3	5	9	4

BubbleSort - Ordenação

1	9	4	5	2	3
1	4	9	5	2	3
1	4	9	5	2	3
1	2	9	5	4	3
1	2	4	9	5	3
1	2	3	9	5	4
1	2	3	5	9	4

BubbleSort - Ordenação

1	9	4	5	2	3
1	4	9	5	2	3
1	4	9	5	2	3
1	2	9	5	4	3
1	2	4	9	5	3
1	2	3	9	5	4
1	2	3	4	9	5

BubbleSort - Ordenação

1	9	4	5	2	3
1	4	9	5	2	3
1	4	9	5	2	3
1	2	9	5	4	3
1	2	4	9	5	3
1	2	3	9	5	4
1	2	3	4	5	9