

Método BubbleSort

Estrutura de Dados II

Prof Jairo Francisco de Souza

Introdução

Ordenar corresponde ao processo de reorganizar um conjunto de objetos em uma ordem ascendente ou descendente

Consiste em facilitar a recuperação posterior de itens do conjunto ordenado

Exemplo: lista telefônica, biblioteca, geração de relatórios

Introdução

Outras situações:

- Teste de unicidade
- Remoção de duplicatas
- Busca
- Encontrar o i -ésimo maior (ou menor) elemento de uma coleção
- Contagem de frequência

Introdução - Conceitos

Tabela:

- Coleção de itens r_1, r_2, \dots, r_n de tamanho n chamados registros
- Uma chave k_i é associada com cada registro r_i e usualmente é um campo do registro
- Ordenação **interna**
 - Memória Principal
- Ordenação **externa**
 - Memória auxiliar
- **Estabilidade** das ordenações

Introdução – Custo x Benefício da Ordenação

Ordenação e busca

Vale a pena ordenar e depois buscar?

Bom senso do programador

Introdução – Análise de Algoritmos

- Complexidade de algoritmos
 - Taxa na qual o armazenamento ou tempo de execução cresce em função do tamanho dos dados de entrada
- Análise Assintótica
 - À medida em que a entrada cresce, a complexidade do algoritmo proporcionalmente tende a uma função conhecida
 - $O(n^2)$, $O(n \log n)$, $O(\log n)$
- Limite Superior e Limite Inferior
 - $O(f(n))$ e $\Omega(f(n))$
- Limite estrito:
 - $T(n) = \theta(f(n))$ sss $T(n) = O(f(n))$ e $T(n) = \Omega(f(n))$

Introdução – Análise de Algoritmos

Notação	Denominação	Comentário	Exemplo
$O(1)$	Constante	O tempo é independente do tamanho da entrada	Acesso a elemento de arranjo
$O(\log n)$	Logarítmico	A base do logaritmo é irrelevante	Busca binária
$O([\log n]^c)$	<u>Polilogarítmico</u>	Não confunda com $O(\log n^c)$ que é o mesmo que $O(\log n)$	
$O(n)$	Linear		Busca seqüencial
$O(n \log n)$	Linearítmico		Heapsort
$O(n^2)$	Quadrático		<u>Bubblesort</u>
$O(n^c)$	Polinomial	Freqüentemente, quando $c > 3$, o algoritmo é impraticável	
$O(c^n)$	Exponencial	Um algoritmo deste tipo tem utilidade apenas para entradas muito pequenas	
$O(n!)$	Fatorial	Um algoritmo deste tipo é impraticável	
$O(n^n)$	Sem denominação	Um algoritmo deste tipo é impraticável	

Introdução Análise de Algoritmos

- Melhor Caso
 - Propriedade dos dados que resultam no melhor resultado possível
- Pior Caso
 - Propriedade dos dados que resultam no pior resultado possível
- Caso Médio
 - Obtido fazendo uma média do desempenho do algoritmo atuando sobre todos os conjuntos de dados possíveis

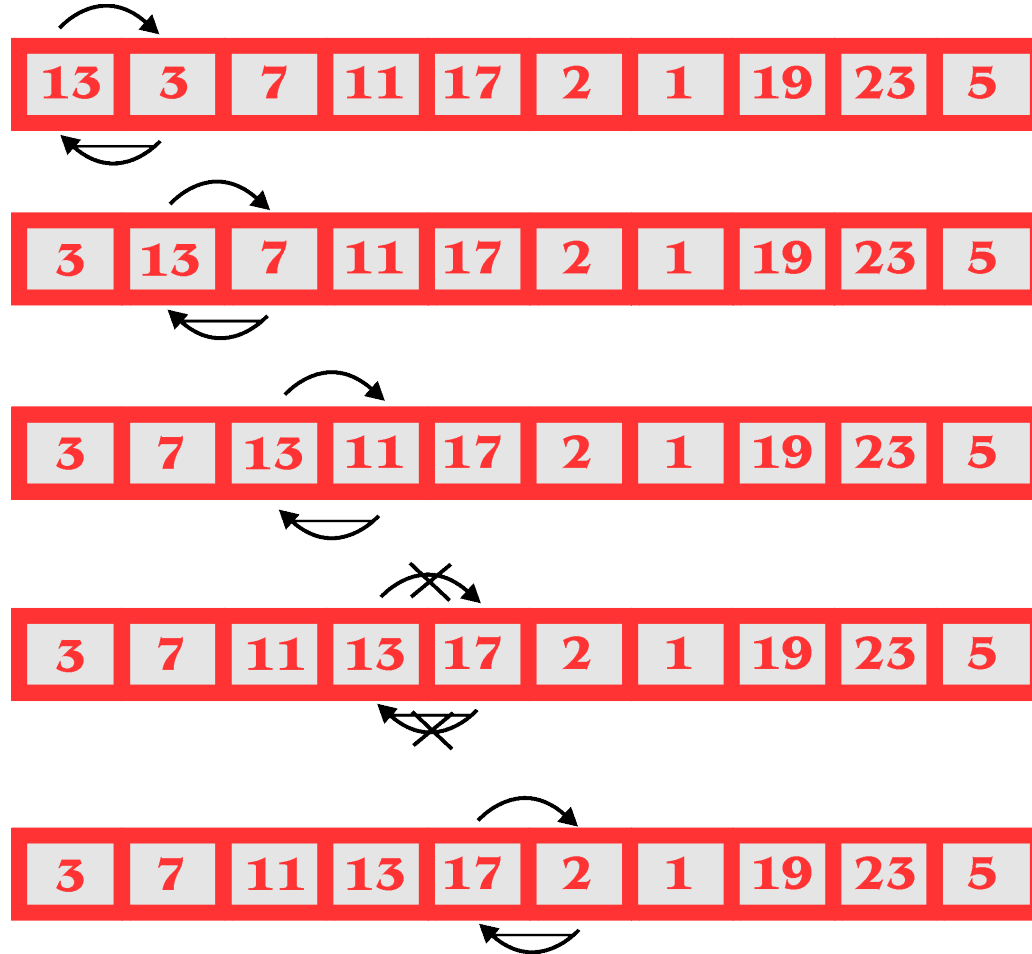
Método BubbleSort (Troca)

- Técnica básica
- Comparam-se dois elementos e trocam-se suas posições se o segundo elemento é menor do que o primeiro
- São feitas várias passagens pela tabela
- Em cada passagem, comparam-se dois elementos adjacentes
- Se estes elementos estiverem fora de ordem, eles são trocados

Método BubbleSort (Troca)

- Vantagens
 - Simplicidade do algoritmo
 - **Estável**
- Desvantagens
 - Lentidão
- Indicações
 - Tabelas muito pequenas
 - Quando se sabe que a tabela está quase ordenada
 - Demonstrações didáticas
- Origem da denominação
 - Os elementos menores (mais “leves”) vão aos poucos “subindo” para o início da tabela, como se fossem bolhas

Método BubbleSort (Troca)



Bubblesort (Ordenação por Troca)

- O algoritmo pode ser descrito em *pseudocódigo*.

Onde:

1. V é um VETOR de elementos que podem ser comparados
2. n é o tamanho desse vetor.

Bubblesort (Ordenação por Troca)

```
(1) procedimento BubbleSort(A : tabela, N: inteiro)
(2)   para j → 1 até N-1 faça
(3)     para i → 1 até N-1 faça
(4)       se A[i] > A[i+1] então
(5)         aux → A[i];
(6)         A[i] → A[i+1];
(7)         A[i+1] → aux;
(8)       fim-se
(9)     fim-para
(10)  fim-para
```

Método BubbleSort (Troca)



44	55	12	42	94	18	06	67
----	----	----	----	----	----	----	----

1ª Iteração (Passo 1):

$$I = 1$$

$$I+1 = 2$$

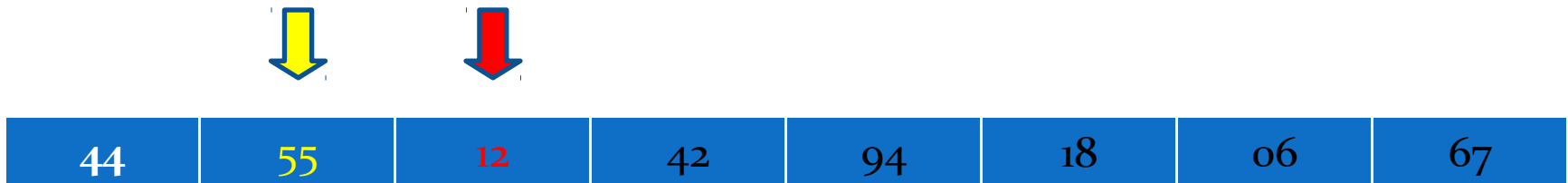
$$A[I] = 44$$

$$A[I+1] = 55$$

Não há troca de elementos

44	55	12	42	94	18	06	67
----	----	----	----	----	----	----	----

Método BubbleSort (Troca)



2ª Iteração (Passo 1):

$I = 2$

$I+1 = 3$

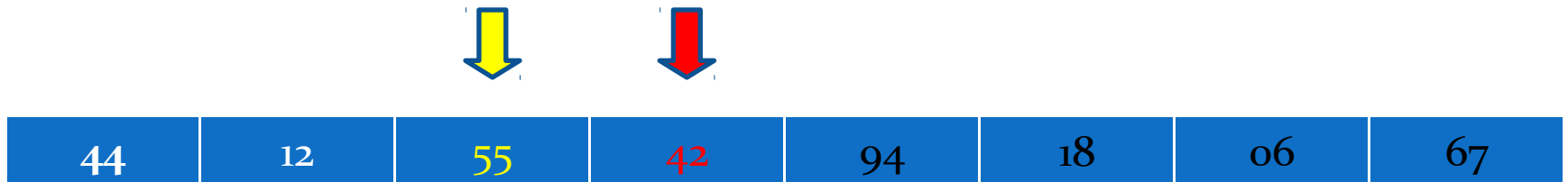
$A[I] = 55$

$A[I+1] = 12$

Há troca de elementos



Método BubbleSort (Troca)



3ª Iteração (Passo 1):

$$I = 3$$

$$I+1 = 4$$

$$A[I] = 55$$

$$A[I+1] = 42$$

Há troca de elementos



Método BubbleSort (Troca)



44	12	42	55	94	18	06	67
----	----	----	----	----	----	----	----

4ª Iteração (Passo 1):

$I = 4$

$I+1 = 5$

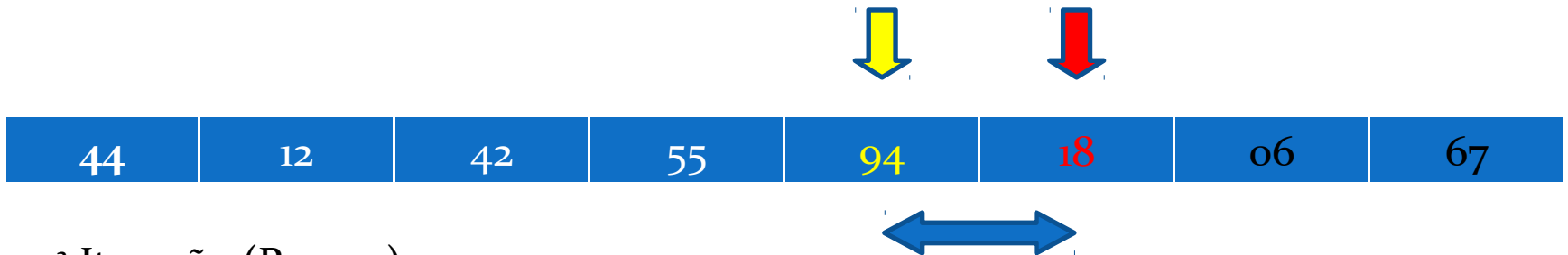
$A[I] = 55$

$A[I+1] = 94$

Não há troca de elementos

44	12	42	55	94	18	06	67
----	----	----	----	----	----	----	----

Método BubbleSort (Troca)



5ª Iteração (Passo 1):

$I = 5$

$I+1 = 6$

$A[I] = 94$

$A[I+1] = 18$

Há troca de elementos

44	12	42	55	18	94	06	67
----	----	----	----	----	----	----	----

Método BubbleSort (Troca)



6ª Iteração (Passo 1):

$I = 6$

$I+1 = 7$

$A[I] = 94$

$A[I+1] = 06$

Há troca de elementos



Método BubbleSort (Troca)



7ª Iteração (Passo 1):

$I = 7$

$I+1 = 8$

$A[I] = 94$

$A[I+1] = 67$

Há troca de elementos



Método BubbleSort (Troca)



44	12	42	55	18	06	67	94
----	----	----	----	----	----	----	----

8ª Iteração (Passo 1):

$I = 8$

$I+1 = ---$

$A[I] = 94$

$A[I+1] = ---$

Não há troca de elementos, chega-se o final da tabela e inicia-se um novo passo para continuar a ordenação

44	12	42	55	18	06	67	94
----	----	----	----	----	----	----	----

Método BubbleSort (Troca)



44	12	42	55	18	06	67	94
----	----	----	----	----	----	----	----



1ª Iteração (Passo 2):

$I = 1$

$I+1 = 2$

$A[I] = 44$

$A[I+1] = 12$

Há troca de elementos

12	44	42	55	18	06	67	94
----	----	----	----	----	----	----	----

Método BubbleSort (Troca)



2ª Iteração (Passo 2):

$I = 2$

$I+1 = 3$

$A[I] = 44$

$A[I+1] = 42$

Há troca de elementos



Método BubbleSort (Troca)



12	42	44	55	18	06	67	94
----	----	----	----	----	----	----	----

3ª Iteração (Passo 2):

$$I = 3$$

$$I+1 = 4$$

$$A[I] = 44$$

$$A[I+1] = 55$$

Não há troca de elementos

12	42	44	55	18	06	67	94
----	----	----	----	----	----	----	----

Método BubbleSort (Troca)



12	42	44	55	18	06	67	94
----	----	----	----	----	----	----	----



4ª Iteração (Passo 2):

$I = 4$

$I+1 = 5$

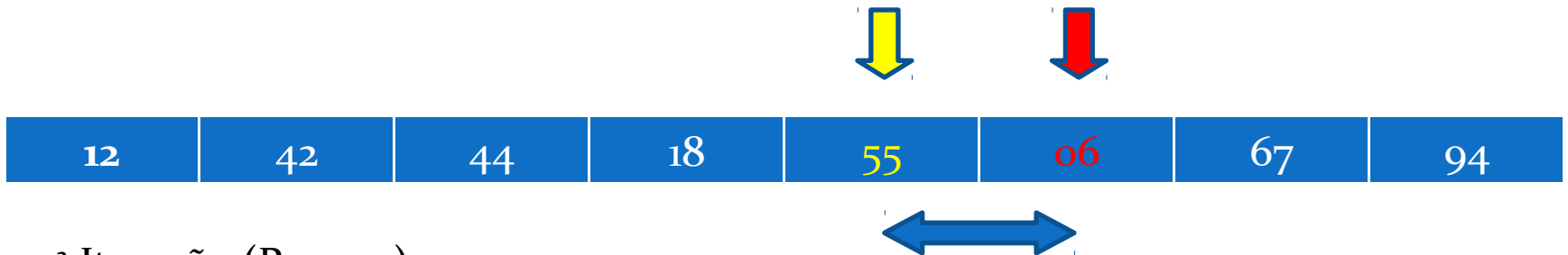
$A[I] = 55$

$A[I+1] = 18$

Há troca de elementos

12	42	44	18	55	06	67	94
----	----	----	----	----	----	----	----

Método BubbleSort (Troca)



5ª Iteração (Passo 2):

$I = 5$

$I+1 = 6$

$A[I] = 55$

$A[I+1] = 06$

Há troca de elementos

12	42	44	18	06	55	67	94
----	----	----	----	----	----	----	----

Método BubbleSort (Troca)

							
12	42	44	18	06	55	67	94

6ª Iteração (Passo 2):

$I = 6$

$I+1 = 7$

$A[I] = 55$

$A[I+1] = 67$

Não há troca de elementos

12	42	44	18	06	55	67	94
----	----	----	----	----	----	----	----

Método BubbleSort (Troca)

							
12	42	44	18	06	55	67	94

7ª Iteração (Passo 2):

$I = 7$

$I+1 = 8$

$A[I] = 67$

$A[I+1] = 94$

Não há troca de elementos

12	42	44	18	06	55	67	94
----	----	----	----	----	----	----	----

Método BubbleSort (Troca)



12	42	44	18	06	55	67	94
----	----	----	----	----	----	----	----

8ª Iteração (Passo 2):

$I = 8$

$I+1 = ---$

$A[I] = 94$

$A[I+1] = ---$

Não há troca de elementos, chega-se o final da tabela e inicia-se um novo passo para continuar a ordenação

12	42	44	18	06	55	67	94
----	----	----	----	----	----	----	----

Método BubbleSort (Troca)



12	42	44	18	06	55	67	94
----	----	----	----	----	----	----	----

1ª Iteração (Passo 3):

$$I = 1$$

$$I+1 = 2$$

$$A[I] = 12$$

$$A[I+1] = 42$$

Não há troca de elementos

12	42	44	18	06	55	67	94
----	----	----	----	----	----	----	----

Método BubbleSort (Troca)



12	42	44	18	06	55	67	94
----	----	----	----	----	----	----	----

2ª Iteração (Passo 3):

$I = 2$

$I+1 = 3$

$A[I] = 42$

$A[I+1] = 44$

Não há troca de elementos

12	42	44	18	06	55	67	94
----	----	----	----	----	----	----	----

Método BubbleSort (Troca)



3ª Iteração (Passo 3):

$I = 3$

$I+1 = 4$

$A[I] = 44$

$A[I+1] = 18$

Há troca de elementos



Método BubbleSort (Troca)



12	42	18	44	06	55	67	94
----	----	----	----	----	----	----	----



4ª Iteração (Passo 3):

$I = 4$

$I+1 = 5$


$A[I] = 44$

$A[I+1] = 06$

Há troca de elementos

12	42	18	06	44	55	67	94
----	----	----	----	----	----	----	----

Método BubbleSort (Troca)



12	42	18	06	44	55	67	94
----	----	----	----	----	----	----	----

5ª Iteração (Passo 3): Não há troca



12	42	18	06	44	55	67	94
----	----	----	----	----	----	----	----

6ª Iteração (Passo 3): Não há troca



12	42	18	06	44	55	67	94
----	----	----	----	----	----	----	----

7ª Iteração (Passo 3): Não há troca

Método BubbleSort (Troca)

- Complexidade $O(n^2)$ onde n é a entrada do problema (quantidade de registros a serem ordenados)

Nº de Chaves	Aleatório	Ordenado	Invertido	Tempo (ms)	C(n)
100	x			16	9801
100		x		0	9801
100			x	16	9801
400	x			31	159201
400		x		31	159201
400			x	32	159201
500	x			62	249001
500		x		47	249001
500			x	63	249001

- Algoritmo estável