

**Algoritmos I**

# Pesquisa e Classificação

*Prof.<sup>a</sup> DSc. Vanessa de Oliveira Campos*

# Introdução

- Duas operações sobre dados são frequentemente utilizadas na solução de problemas: pesquisa e classificação de vetores.
  - **Pesquisa**: envolve examinar um conjunto de dados *procurando* determinar a presença (ou ausência) de um ou mais valores no mesmo e, se presentes, sua localização.
  - **Classificação** (ou ordenação): envolve rearranjar um conjunto de dados segundo alguma ordem predeterminada.
- Boas soluções para essas duas operações se tornam importantes à medida que aumenta o volume dos dados a processar.



# Pesquisa Sequencial

- Trata-se do método de pesquisa mais simples e intuitivo.
- Consiste em examinar o conjunto de dados, **valor a valor**, do primeiro ao último, verificando se o valor procurado está presente.
- O processo termina no momento em que o valor for encontrado ou, caso não seja encontrado, ao ser atingido o fim do vetor.



# Pesquisa Sequencial

- Não exige que os dados estejam ordenados.
- É interessante apenas para conjuntos relativamente pequenos de dados ou em situações em que o número de pesquisas é muito pequeno.
- O pior caso dessa estratégia é quando o valor não existe no conjunto, pois todos os elementos têm que ser examinados.
- Na média, a metade dos valores deve ser examinada.



# Pesquisa Binária

- Exige que o conjunto de dados esteja ordenado.
- O método consiste na divisão sucessiva do conjunto de dados pela metade.
- A cada divisão, a partir da primeira, compara-se o valor procurado com o valor que se encontra no ponto médio determinado pela divisão.
  - Se esse valor é o procurado, a pesquisa é encerrada.
  - Se não for, verifica-se se o valor que se está buscando é maior ou menor do que o valor do termo médio e determina-se qual a metade dos dados que ainda tem potencial para conter o valor.



# Pesquisa Binária

- Exemplo:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
002	015	017	030	032	034	040	050	080	090	093	097	099	101	101

**Valor buscado: 32**



# Pesquisa Binária

- Exemplo:

<div>1</div> <div></div>														
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
002	015	017	030	032	034	040	050	080	090	093	097	099	101	101

**Valor buscado: 32**

inf	sup	med
0	14	7



# Pesquisa Binária

■ Exemplo:

			2				1							
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
002	015	017	030	032	034	040	050	080	090	093	097	099	101	101

**Valor buscado: 32**

inf	sup	med
0	14	7
0	6	3





# Pesquisa Binária

■ Exemplo:

			2		3		1							
			↓		↓		↓							
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
002	015	017	030	032	034	040	050	080	090	093	097	099	101	101

**Valor buscado: 32**

inf	sup	med
0	14	7
0	6	3
4	6	5



# Pesquisa Binária

■ Exemplo:

			2	4	3		1							
			↓	↓	↓		↓							
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
002	015	017	030	032	034	040	050	080	090	093	097	099	101	101

**Valor buscado: 32**

inf	sup	med
0	14	7
0	6	3
4	6	5
4	4	4

**Achou!**



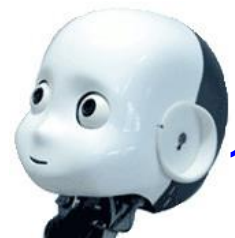
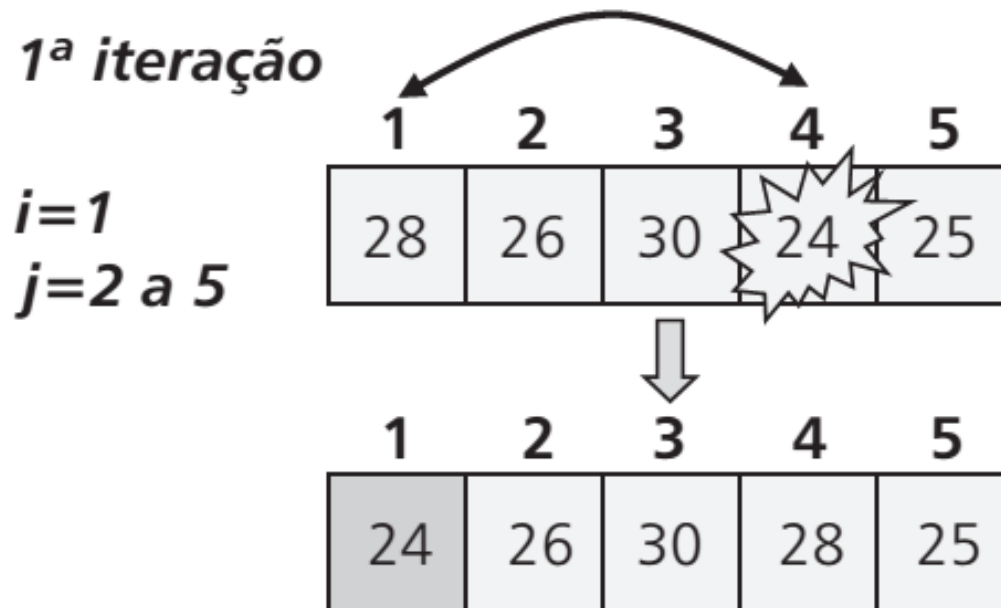
# Classificação por Seleção

- Para colocar os dados em ordem crescente, percorre-se o vetor buscando o elemento com o menor valor.
  - Esse é trocado com aquele que está na primeira posição do vetor.
- O método é, então, repetido para o segmento do vetor que inicia na segunda posição, e assim sucessivamente.



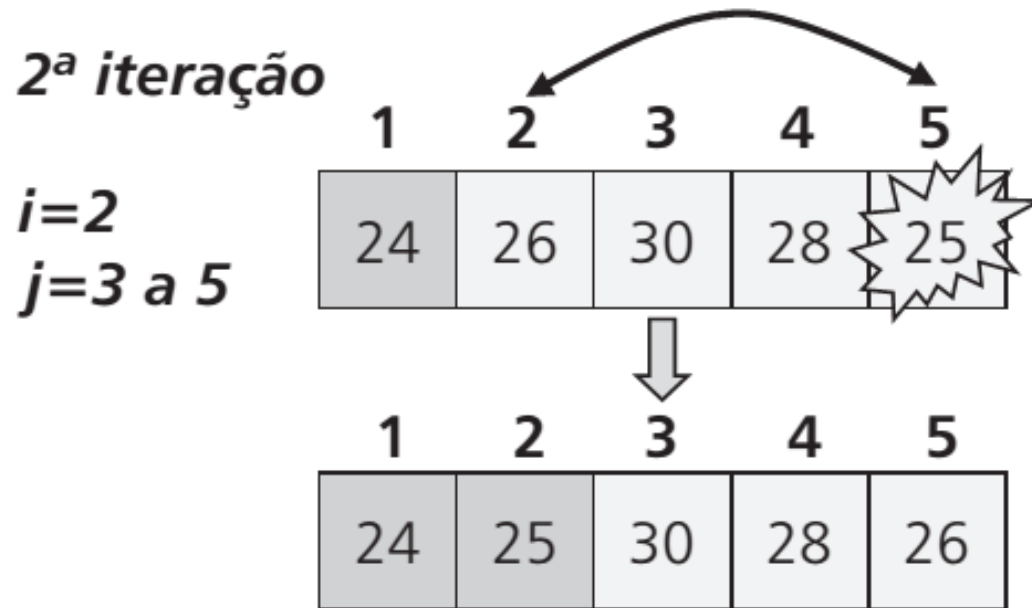
# Classificação por Seleção

- Exemplo:



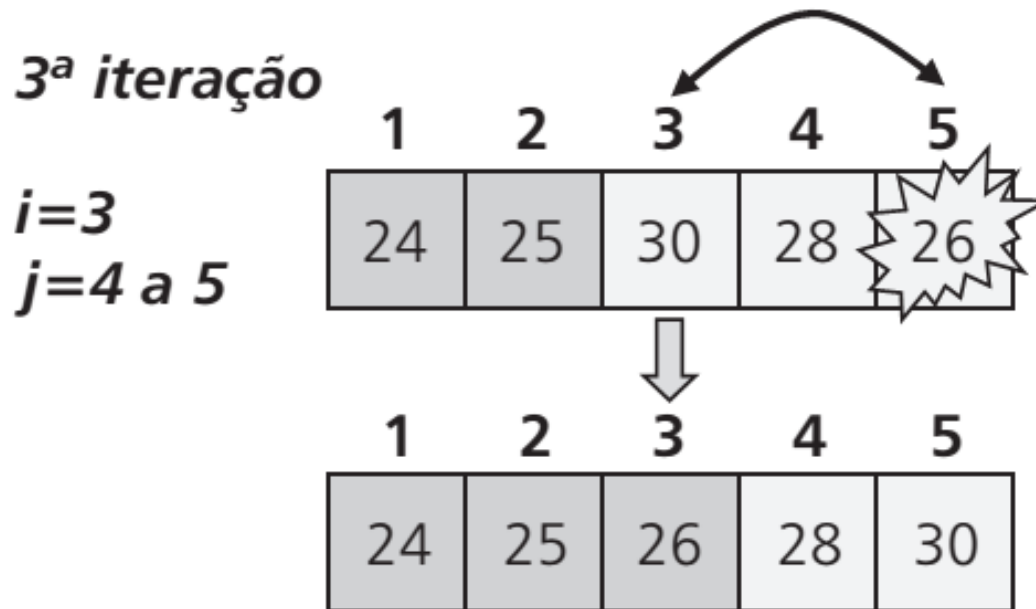
# Classificação por Seleção

- Exemplo:



# Classificação por Seleção

- Exemplo:



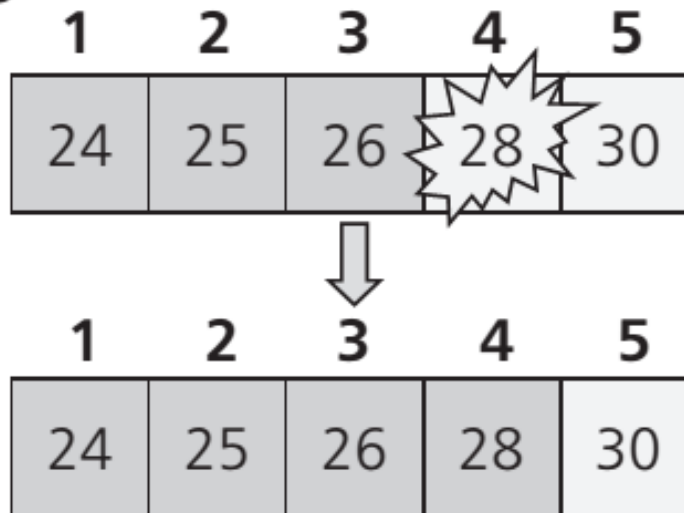
# Classificação por Seleção

- Exemplo:

*4ª iteração*

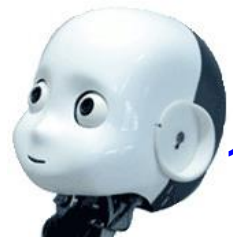
*$i=4$*

*$j=5$  a  $5$*



# Classificação – Método da Bolha (Bubble Sort)

- Consiste em percorrer o vetor comparando cada dois elementos adjacentes e trocando-os de posição caso estejam em ordem contrária à desejada.
- É feito um controle, a cada varredura do vetor, para saber se houve alguma troca de valores e, se houve, nova varredura é executada.
- O ponto onde a última troca aconteceu é fixado como o limite da próxima varredura, reduzindo progressivamente o número de elementos do conjunto considerado na varredura.





# Método da Bolha

- Exemplo:

*1ª iteração*



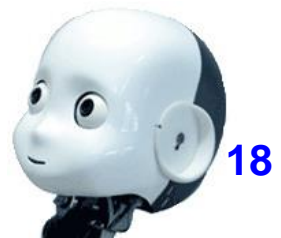
# Método da Bolha

- Exemplo:

*1ª iteração*

28	26	30	24	25
----	----	----	----	----

26	28	30	24	25
----	----	----	----	----



# Método da Bolha

- Exemplo:

*1ª iteração*

28	26	30	24	25
----	----	----	----	----

26	28	30	24	25
----	----	----	----	----

26	28	30	24	25
----	----	----	----	----



# Método da Bolha

- Exemplo:

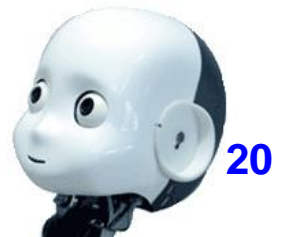
*1ª iteração*

28	26	30	24	25
----	----	----	----	----

26	28	30	24	25
----	----	----	----	----

26	28	30	24	25
----	----	----	----	----

26	28	24	30	25
----	----	----	----	----



# Método da Bolha

- Exemplo:

*1ª iteração*

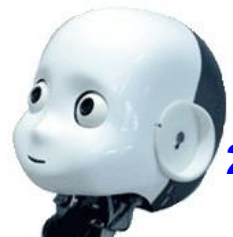
28	26	30	24	25
----	----	----	----	----

26	28	30	24	25
----	----	----	----	----

26	28	30	24	25
----	----	----	----	----

26	28	24	30	25
----	----	----	----	----

26	28	24	25	30
----	----	----	----	----

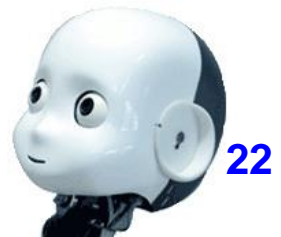


# Método da Bolha

- Exemplo:

*2ª iteração*

26	28	24	25	30
----	----	----	----	----



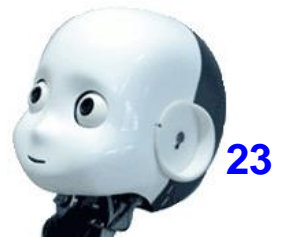
# Método da Bolha

- Exemplo:

*2ª iteração*

26	28	24	25	30
----	----	----	----	----

26	28	24	25	30
----	----	----	----	----



# Método da Bolha

- Exemplo:

*2ª iteração*

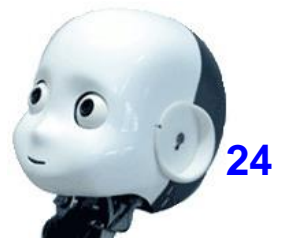
26	28	24	25	30
----	----	----	----	----

26	28	24	25	30
----	----	----	----	----

Diagram showing a swap between 28 and 24 in the second iteration.

26	24	28	25	30
----	----	----	----	----

Diagram showing a swap between 24 and 28 in the second iteration.





# Método da Bolha

- Exemplo:

*2ª iteração*

26	28	24	25	30
----	----	----	----	----

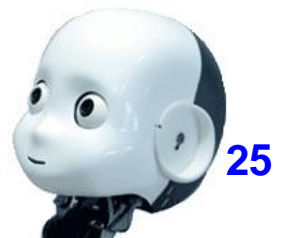
26	28	24	25	30
----	----	----	----	----

Diagram showing a swap between 28 and 24 in the second iteration.

26	24	28	25	30
----	----	----	----	----

Diagram showing a swap between 24 and 28 in the second iteration.

26	24	25	28	30
----	----	----	----	----



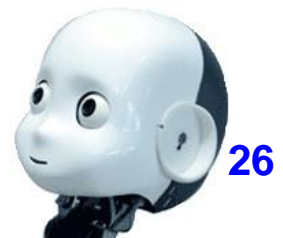
# Método da Bolha

- Exemplo:

*3ª iteração*



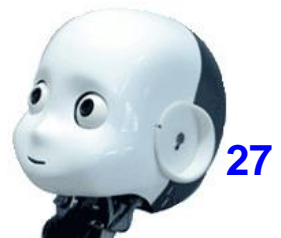
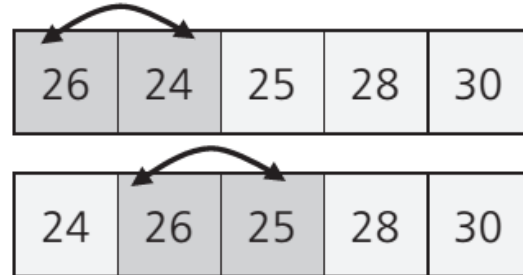
Para ignorar o último que já está obrigatoriamente ordenado, basta comocar  $n - 1$



# Método da Bolha

- Exemplo:

*3ª iteração*



# Método da Bolha

- Exemplo:

*3ª iteração*

26	24	25	28	30
----	----	----	----	----

24	26	25	28	30
----	----	----	----	----

24	25	26	28	30
----	----	----	----	----

