CLASSIFICAÇÃO DE DADOS

Classificação (Sorting)

- Processo de organizar itens em ordem (de)crescente, segundo algum critério
- Também chamado de ordenação
- Aplicações de *Sorting*
  - Preparação de dados para facilitar pesquisas futuras
    - Exemplo: dicionários e listas telefônicas
  - Agrupar itens que apresentam mesmos valores
    - Para eliminação de elementos repetidos
  - Batimento entre itens presentes em mais de um arquivo
    - Para combinação de dados presentes nos vários arquivos
    - Para consolidação dos vários arquivos em um único

Definições

- Sejam $R_1, R_2, \ldots, R_N$, N itens (chamados registros)
- Cada registro R_i, é formado por uma chave C_i e por informações ditas satélites
- A ordenação dos registros é feita definindo-se uma relação de ordem "<" sobre os valores das chaves
- O objetivo da ordenação é determinar uma permutação dos índices $1 \leq i_1, i_2, \ldots, i_N \leq N$ das chaves, tal que $C_{i1} \leq C_{i2} \leq \ldots \leq C_{iN}$.
- Um conjunto de registros é chamado de arquivo

Relação de Ordem

- Uma relação de ordem "**<**" (leia-se precede) deve satisfazer as seguintes condições para quaisquer valores ***a***, ***b*** e ***c***:

  (i) Uma e somente uma das seguintes possibilidades é verdadeira: $a < b$, $a = b$ ou $b < a$ (lei da tricotomia)

  (ii) Se $a < b$ e $b < c$, então $a < c$ (transitividade)

- As propriedades (i) e (ii) definem o conceito de ordem linear ou ordem total

Mais Definições

- Um algoritmo de classificação é dito estável, se ele preserva a ordem relativa original dos registros com mesmo valor de chave.

- **Pergunta:** Qual a importância prática de algoritmos de classificação estável?

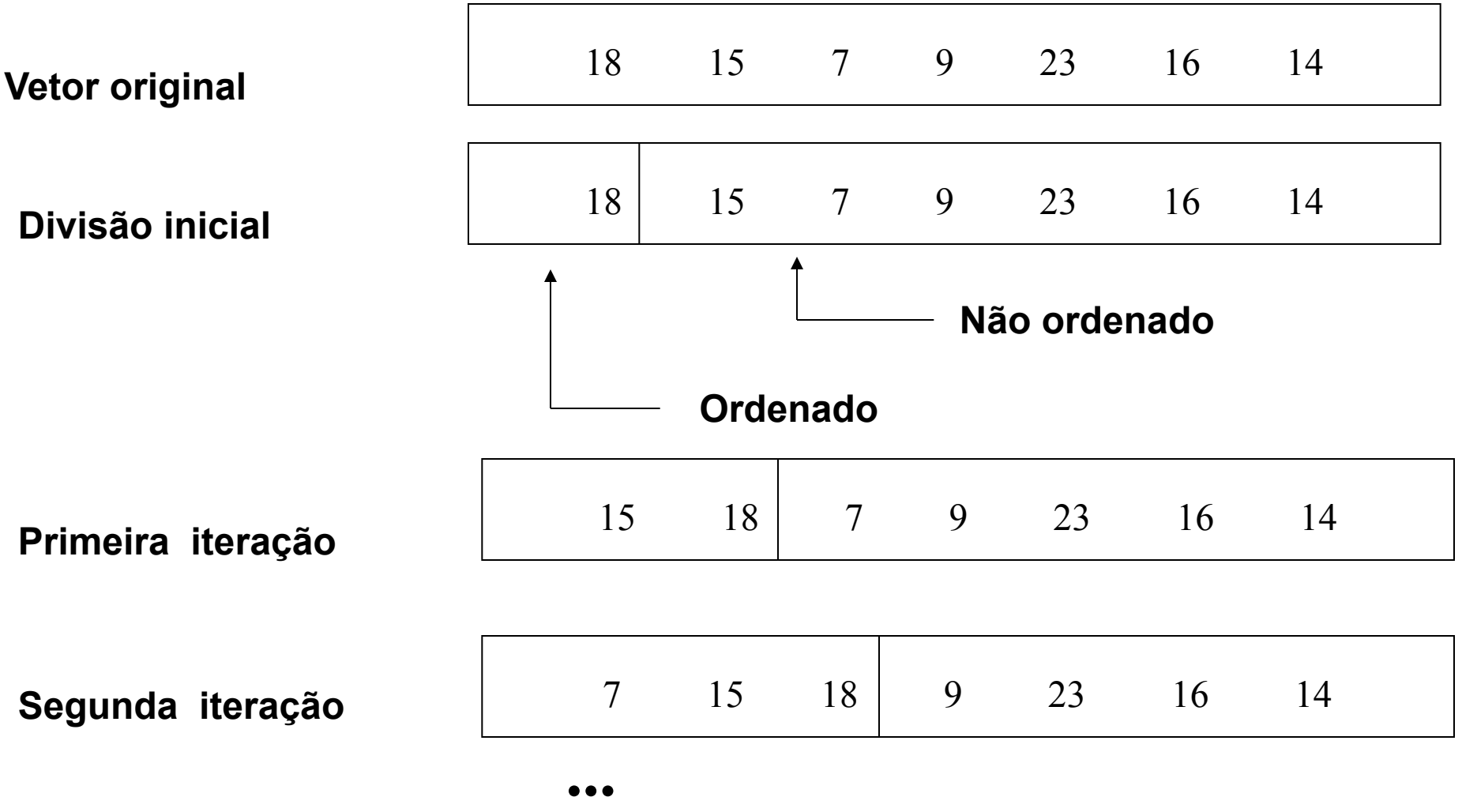
Métodos de Classificação de Dados

- Inserção Direta (*Insertion Sort*)
- Seleção Direta (*Selection Sort*)
- BubbleSort

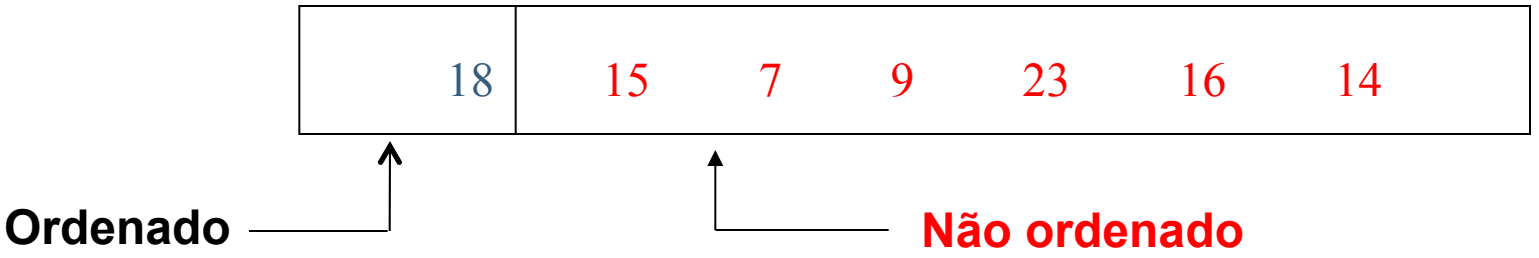
Inserção Direta

- Mais simples
- Normalmente utilizado para um conjunto pequeno de dados, pois apresenta baixa eficiência
- Divide o vetor em 2 segmentos:
  - o primeiro contendo os elementos já ordenados
  - o segundo contendo os elementos ainda não ordenados
- **Funcionamento**: Pega o primeiro elemento do segmento não ordenado e procura seu lugar no segmento ordenado.
- No início: o 1° segmento terá apenas 1 elemento

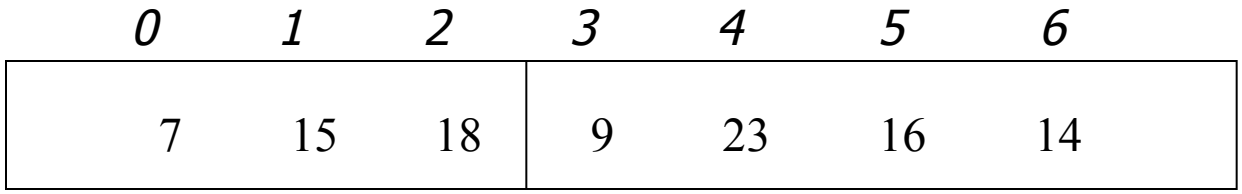
Classificação de Dados
Método da Inserção Direta



Inserção direta



- [15, 18, 7, 9, 23, 16, 14] após 1ª iteração
- [7, 15, 18, 9, 23, 16, 14] após 2ª iteração
- [7, 9, 15, 18, 23, 16, 14] após 3ª iteração
- [7, 9, 15, 18, 23, 16, 14] após 4ª iteração
- [7, 9, 15, 16, 18, 23, 14] após 5ª iteração
- [7, 9, 14, 15, 16, 18, 23] após 6ª iteração




```
public void insertionSort (int[] a) {
    for (int i = 1; i < a.length; i++) {
        int j = i;  // pos do 1° elemento no seg. não ord.
        int B = a[i]; // 1° elemento no seg. não ord.
        while ((j > 0) && (a[j-1] > B)) {
            a[j] = a[j-1];
            j--;
        }
        a[j] = B;
    }
}
```

buscando a posição do 1º elemento do segmento não ordenado no segmento ordenado

Bubblesort (Método da Bolha)

Exemplo: Suponha que se deseja classificar em ordem crescente o seguinte vetor de chaves:

28	26	30	24	25

Primeira Varredura

28	26	30	24	25	compara par (28, 26) : troca
26	28	30	24	25	compara par (28, 30) : não troca
26	28	30	24	25	compara par (30, 24) : troca
26	28	24	30	25	compara par (30, 25) : troca
26	28	24	25	30	fim da primeira varredura

Bubblesort (Método da Bolha)

Segunda Varredura

26 28 24 25 30	compara par (26, 28) : não troca
26 28 24 25 30	compara par (28, 24) : troca
26 24 28 25 30	compara par (28, 25) : troca
26 24 25 28 30	fim da segunda varredura

Terceira Varredura

26 24 25 28 30	compara par (26, 24) : troca
24 26 25 28 30	compara par (26, 25) : troca
24 25 26 28 30	fim da terceira varredura

Classificação de Dados
Método da Bolha - Bubblesort

Considerando o seguinte vetor :

13	11	25	10	18	21	23

1) Em quantas varreduras o vetor é classificado ?

2) Como identificar, a partir do final de cada varredura, quantos elementos já estão classificados?

Observe que a quantidade de elementos, a partir do último, que pode ser ignorada de uma varredura para a outra é conhecida pela posição na qual ocorreu a última troca.
A partir daquele ponto, o vetor já se encontra classificado!

Classificação de Dados
Método da Bolha - Bubblesort


```
public void bubbleSort(int[] a) {
        int i = a.length-1;
        while (i > 0) {
                int lastFlipped = 0;
                for (int j = 0; j < i; j++) {
                        if (a[j] > a[j+1]) { // troca par de posição
                                int T = a[j];
                                a[j] = a[j+1];
                                a[j+1] = T;
                                lastFlipped = j;
                        }
                }
                i = lastFlipped;
        }
}
```

Seleção direta

- Princípio de classificação
  - a seleção do menor elemento é feita por pesquisa seqüencial
  - o menor elemento encontrado é permutado com o que ocupa a posição inicial do vetor, que fica reduzido de um elemento
  - o processo de seleção é repetido para a parte restante do vetor, até que todas os elementos tenham sido selecionados e colocados em suas posições definitivas

Método da seleção direta

Exercício:

Suponha que se deseja classificar o seguinte vetor:

9 25 10 18 5 7 15 3

Simule as iterações necessárias para a classificação.

Método da seleção direta

Iteração	Vetor	Chave Selecionada	Permutação	Vetor ordenado até a posição
1	9 25 10 18 5 7 15 3	3	9 e 3	
2	3 25 10 18 5 7 15 9	5	25 e 5	0
3	3 5 10 18 25 7 15 9	7	10 e 7	1
4	3 5 7 18 25 10 15 9	9	18 e 9	2
5	3 5 7 9 25 10 15 18	10	25 e 10	3
6	3 5 7 9 10 25 15 18	15	25 e 15	4
7	3 5 7 9 10 15 25 18	18	25 e 18	5
8	3 5 7 9 10 15 18 25			6

Método da seleção direta


```
public static void selectionSort(int[] a) {
        int min = 0;
        for (int i = 0; i < a.length-1; i++) {
                min = i; // mínimo inicial
                for (int j = i+1; j < a.length; j++)
                        if (a[j] < a[min])
                                min = j;  // acha o novo mínimo
                int T = a[i];  // coloca o novo mínimo (min)
                a[i] = a[min]; // na posição correta (i)
                a[min] = T;
        }
}
```