

## Ordenação de Vetor

SSC300- Linguagens de Programação e Aplicações

## Ordenação

- ORDENAÇÃO é uma das **tarefas básicas** em processamento de dados
- A **ordenação de um vetor** significa classificar os seus elementos
  - Colocar os elementos do vetor em ordem de acordo com algum critério



2

## Ordenação de Vetores

- Supor que os elementos de um vetor sejam **inteiros**
- Critério de ordenação:
  - ordem **crecente**
  - ordem **decrecente**



3

## Algoritmos de Ordenação

- Existem **vários algoritmos** de ordenação
  - Diferenciam pela velocidade da ordenação (e complexidade de implementação)
- Métodos de ordenação: (simples)
  - **Método da Seleção**
  - **Método da Bolha**



4

## Ordenação pelo Método da Seleção



## Método da Seleção



- Funcionamento:
  - a cada passagem pelo vetor, selecionar o **menor** elemento e colocar este elemento o mais a **esquerda possível** no vetor.
- Para isto deve-se **trocar as posições** dos elementos do vetor
  - Na **primeira** passagem troca-se o menor elemento com o que está na primeira posição
  - Na **segunda** passagem troca-se o segundo menor elemento com o que está na segunda posição
  - Assim por diante

6

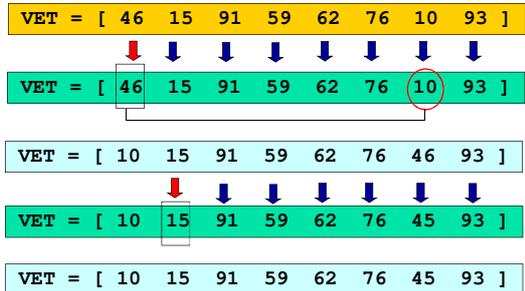
## Método da Seleção



- Desse modo, na **passagem i** só se deve procurar o **menor** elemento a partir do **i-ésimo** até o **N-ésimo** elemento do vetor (os elementos anteriores já estarão ordenados).
- Note-se que serão necessárias **(N-1)** passagens, pois, na última passagem, o N-ésimo elemento já estará na sua posição correta

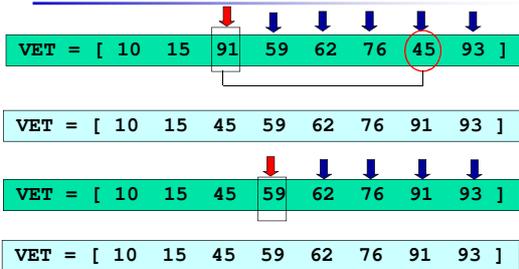
7

## Método da Seleção (ordem crescente)



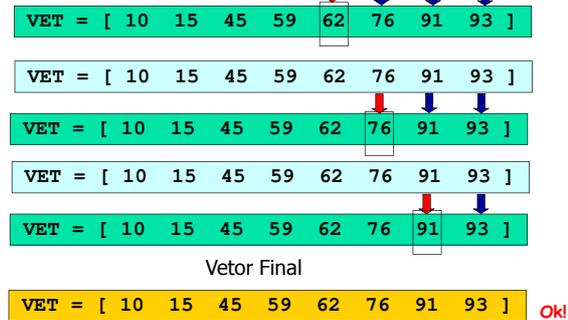
8

## Método da Seleção (ordem crescente)



9

## Método da Seleção (ordem crescente)



Ok!  
10

## Programa de Ordenção - Método da Seleção

```
void ordena_selecao(int vetor[], int tam)
{
    int i, j, min, aux;
    for(i=0; i < tam; i++) {
        min = i;
        for(j= i+1; j < tam; j++) {
            if (vetor[j] < vetor[min])
                min = j;
        }
        aux=vetor[j];
        vetor[j]=vetor[min];
        vetor[min]=aux;
    }
}
```

O vetor é passado como parâmetro – por referência

Percorre todo o vetor até o elemento n-1

Percorre do elemento i+1 até o último. Armazena a posição (índice) do menor elemento

Troca os elementos, colocando-os na ordem crescente

A variável *i* controla o pivô

A variável *j* controla os elementos a frente do pivô

11

## Exercício de Fixação

- Utilizando o algoritmo anterior (método da seleção), faça a ordenação do seguinte vetor:
  - Vetor = { 4, 8, 2, 3, 7}



12

# Método da Bolha




# Método da Bolha



- É mais popular que o método da seleção, embora seu desempenho seja inferior
- Funcionamento:
  - A cada passagem pelo vetor, o elemento da **i-ésima posição** é selecionado e transferido para a sua posição adequada.

14

VETOR DESORDENADO

[ 15	46	91	59	62	76	10	93 ]
[ 15	46	91	59	62	76	10	93 ]
[ 15	46	91	59	62	10	76	93 ]
[ 15	46	91	59	10	62	76	93 ]
[ 15	46	91	10	59	62	76	93 ]
[ 15	46	10	91	59	62	76	93 ]
[ 15	10	46	91	59	62	76	93 ]
[ 10	15	46	91	59	62	76	93 ]
[ 10	15	46	59	91	62	76	93 ]
[ 10	15	46	59	62	91	76	93 ]
[ 10	15	46	59	62	76	91	93 ]

VETOR ORDENADO



15

# Programa Ordenação - Método da Bolha



```

void ordena_bolha(int vetor[], int tam)
{
    int i, j, aux;
    for(i=0; i < tam; i++)
        for(j=tam-1; j > i; j--)
            if(vetor[j-1] > vetor[j]) {
                aux=vetor[j-1];
                vetor[j-1]=vetor[j];
                vetor[j]=aux;
            }
}

```

16

## Exercício de Fixação

- Utilizando o algoritmo anterior (método da bolha), faça a ordenação do seguinte vetor:
  - Vetor = { 4, 8, 2, 3, 7}



17