

## • Selection Sort (Bubbles)

→ Ordenação por Seleção

→ Ideia:

- a cada passo | procura o menor valor do array e o coloca na primeira posição do array
- descarta-se a primeira posição do array e repete-se para segunda
- Repete-se para todas as posições do array

→ Performance

- melhor caso :  $O(N^2)$
- pior caso :  $O(N^2)$
- caso médio :  $O(N^2)$
- Ineficiente para grandes conjuntos de dados
- Estável: não altera a ordem de dados iguais



## Algoritmo (em C):

1. void selectionSort (int \*V, int n) {

2.     int i, j, menor, aux;

3.     for (i = 0; i < n - 1; i++) {

4.         menor = i;

5.         for (j = i + 1; j < n; j++) {

6.             if (V[j] < V[menor])

7.                 menor = j;

8.         }

9.         if (i != menor) {

10.             aux = V[i];

11.             V[i] = V[menor];

12.             V[menor] = aux;

13.         }

14.     }

15. }

Procura menor elemento em relação a "i"

Troca os valores da posição atual com a "menor"



• Exemplo

23 4 67 -8 90 54 21

Vetor não ordenado

iteração 0:

<sup>i</sup> 23	4	67	<sup>menor</sup> -8	90	54	21	trocar
-8	4	67	23	90	54	21	

iteração 1:

-8	<sup>i, menor</sup> 4	67	23	90	54	21	ok
-8	4	67	23	90	54	21	

iteração 2:

-8	4	<sup>i</sup> 67	23	90	54	<sup>menor</sup> 21	Trocar
-8	4	21	23	90	54	67	

iteração 3:

-8	4	21	<sup>i, menor</sup> 23	90	54	67	ok
----	---	----	---------------------------	----	----	----	----



iteração 4:

				i	menor	
-8	4	21	23	90	54	Trocar
-8	4	21	23	54	90	67

iteração 5:

					i	menor	
-8	4	21	23	54	90	67	Trocar
-8	4	21	23	54	67	90	

• Vektor ordenado:

-8 4 21 23 54 67 90

• Pseudo código

Selection Sort ( $V, \text{tam}$ )

→ Para cada posição  $N$  entre 0 e  $\text{tam}-1$ , faça:

- encontre o índice  $i$  que contém o menor valor entre  $N$  e  $\text{tam}-1$

- se o menor valor é diferente do valor da posição  $N$

\* troque os valores de  $V[\text{menor}]$  e  $V[N]$



## • Observações Gerais:

\* Vantagens: não altera a ordem dos dados iguais (estável)

\* Desvantagens:

- sua eficiência diminui drasticamente a medida que o número de elementos no array aumenta

- Não é recomendado para aplicações que envolvam grandes quantidades de dados ou que precisem de velocidade

\* Complexidade

- sempre na ordem  $O(N^2)$ , não depende da ordem inicial dos elementos

- melhor que bubble sort, envolve um menor número de comparações