

Ordenação

por inserção e seleção

Isadora Yasmim da Silva
Matrícula: 202403076

Ordenação por inserção

Insertion sort

Como funciona?

Dado um vetor preenchido numericamente é feita uma comparação entre os dois primeiros termos da lista, se o 2º for maior que o primeiro sua posição é invertida com o 1º, senão as posições são mantidas e uma “nova lista” será formada:

[4,7,2,5,4,0] --> $7 > 4$ --> sim --> [4,7]

Após isso cada novo elemento deve ser comparado com os da “nova lista”:

[4,7,2,5,4,0] --> $2 > 7$ --> não --> [4,2,7]

[4,7,2,5,4,0] --> $2 > 4$ --> não --> [2,4,7]

O elemento “2” foi **inserido** na nova lista de forma **ordenada**

InsertionSort(vetor)

Seja n o tamanho de vetor

Para $i = 1$ até n

 chave = vetor[i]

$j = i - 1$

 enquanto $j \geq 0$ e $\text{vetor}[j] > \text{chave}$

 vetor[j + 1] = vetor[j]

$j = j - 1$

 vetor[j + 1] = chave

```
package main
```

```
import "fmt"
```

```
func InsertionSort(vetor []int) {
```

```
    n := len(vetor)
```

```
    for i := 1; i < n; i++ {
```

```
        chave := vetor[i]
```

```
        j := i - 1
```

```
        // Move os elementos de vetor que são maiores que a chave para  
        posição seguinte
```

```
        for j >= 0 && vetor[j] > chave {
```

```
            vetor[j+1] = vetor[j]
```

```
            j = j - 1
```

```
        }
```

```
        vetor[j+1] = chave
```

```
    }
```

```
}
```

```
func main() {
```

```
    vetor := []int{4, 7, 2, 5, 4, 0}
```

```
    fmt.Println("Vetor original:", vetor)
```

```
    InsertionSort(vetor)
```

```
    fmt.Println("Vetor ordenado:", vetor)
```

```
}
```

Ordenação por seleção

Selection sort

Como funciona?

Dado um vetor preenchido numericamente é feito a seleção de seu menor elemento e o posicionamento desse elemento de menor índice disponível na “nova lista”:

[4,7,2,5,4,0] --> 0 --> [0]

O elemento “0” foi inserido na “nova lista” de forma ordenada e retirado da “lista original”. Assim é necessário repetir o processo até que a ordenação seja concluída:

[4,7,2,5,4] --> 2 --> [0,2]

[4,7,5,4] --> 4 --> [0,2,4]

[7,5,4] --> 4 --> [0,2,4,4]

[7,5] --> 5 --> [0,2,4,4,5]

[7] --> 7 --> [0,2,4,4,5,7]

SelectionSort(vetor)

Seja n o tamanho de vetor

Para $i = 1$ até $n - 1$

 menorindice = i

 para $j = i + 1$ até n

 se $\text{vetor}[j] < \text{vetor}[\text{menorindice}]$

 menorindice = j

 fimse

//inverte os valores dos elementos

temp = $\text{vetor}[\text{menorindice}]$

$\text{vetor}[\text{menorindice}] = \text{vetor}[i]$

$\text{vetor}[i] = \text{temp}$

```
package main
```

```
import "fmt"
```

```
func SelectionSort(vetor []int) {  
    n := len(vetor)  
    for i := 0; i < n-1; i++ {  
        menorIndice := i  
        for j := i + 1; j < n; j++ {  
            if vetor[j] < vetor[menorIndice] {  
                menorIndice = j  
            }  
        }  
  
        temp := vetor[menorIndice]  
        vetor[menorIndice] = vetor[i]  
        vetor[i] = temp  
    }  
}
```

```
func main() {  
    vetor := []int{4, 7, 2, 5, 4, 0}  
    fmt.Println("Vetor original:", vetor)  
  
    SelectionSort(vetor)  
  
    fmt.Println("Vetor ordenado:", vetor)  
}
```

Referências bibliográficas

- INSERTION SORT | Algoritmos #4
- Algoritmo SELECTION SORT | Algoritmos de Ordenação | Algoritmos #3
- Madeira, H. Félix, N. Algoritmos de Ordenação