



Ana Luísa

Algoritmos de Ordenação

Ana Luísa SILVA Gonçalves 202403059

Ordenação por inserção

- Método Simples - usado em vetores menores
- Estável: a ordem relativa dos elementos iguais permanece a mesma durante toda a ordenação
- A lista é construída aos poucos

Como funciona?

- Os elementos são separados em duas partes: a dos elementos já organizados, e a dos elementos ainda não organizados;
- A primeira posição é sempre considerada como ordenada;
- Então a lista desordenada vai sendo percorrida e os elementos vão sendo colocados em suas posições corretas na lista ordenada;
- Esse processo se repete até que toda a lista esteja organizada.

Algoritmo

```
11 Inicio
12   Receba um vetor com k elementos
13   Para i=1 até k-1 faça
14     atual = vetor[i]
15     j = i-1
16     Enquanto j > 0 e vetor [j]>atual faça
17       vetor[j + 1] = vetor[j]
18       j = j - 1
19     fim-enquanto
20     A[j+1] = atual
21   fim-para
22   Retorne o vetor k ordenado
```

Código em GO

```
12 package main
13 import "fmt"
14 func InsertionSort() {
15     for i := 1; i < n; i++ {
16         tmp := array[i]
17         j := i - 1
18
19         for (j >= 0) && (array[j] > tmp) {
20             array[j+1] = array[j]
21             j--
22         }
23         array[j+1] = tmp
24     }
25 }
26 }
27 }
```

Ordenação por seleção

- Método Simples
- Não é estável

Como funciona?

- Da mesma forma que acontece na ordenação por inserção, na ordenação por seleção a lista a ser ordenada também é dividida em duas partes: a parte dos elementos já ordenados, e a parte dos que serão ordenados;
- A lista dos ordenados começa vazia até que a lista seja percorrida e o menor elemento é colocado na primeira posição.
- A cada iteração a lista é percorrida em busca do menor elemento, que vai sendo colocado á direita do menor elemento encontrado antes dele;
- Dessa forma, ao longo do processo, a sub-lista dos itens desordenados diminui, até que não sobre nenhum, enquanto a sub-lista dos itens ordenados vai se completando.

Algoritmo

```
Receba um vetor com k elementos
Para i=0 até n-2 faça
  Inicie min=i
  Para j = i+1 ate k-1 faça
    Se vetor[j]<vetor[min], então min=j
  fim_para
  Se min != i, então
    troque vetor [i] com vetor [min]
  fim_se
fim_para
Retorne o vetor ordenado
```


Código em GO

```
package main
import "fmt"

func SelectionSort() {
    for i = 0; i < (n - 1); i++ {
        indice := i
        for j = (i + 1); j < n; j++ {
            if array[indice] > array[j] {
                indice = j
            }
        }
        swap(indice, i)
    }
}
```

