

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 permace 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 comsiderada como ordenada;
- Então a lista desordenada vai sendo percorrida e os elementos vão sendo colocados em suas posições corretas na lista ordenada;
- Esse processso se repete até que toda a lista esteja organizada.

Algoritmo

```
| Inicio | Receba um vetor com k elementos | Para i=1 até k-1 faça | 4 | atual = vetor[i] | 15 | j = i-1 | 16 | Enquanto j > 0 e vetor [j]>atual faça | 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

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 desornados dimuinui, até que nãosobre 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-l faça

Se vetor[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)
}
```

Fontes:

https://www.treinaweb.com.br/blog/conheca-os-principais-algoritmos-de-ordenacao#:~:text=Insertion%20Sort%20ou%20ordena%C3%A7%C3%A3o%20por,um%20m%C3%A9todo%20de%20ordena%C3%A7%C3%A3o%20est%C3%A1vel.

https://www.estrategiaconcursos.com.br/blog/algorit mo-ordenacao-por-selecao/

https://github.com/Henriquemcc/Algoritmos_de_or denacao_em_Go/tree/main



Q

HOMEPAGE > CONCURSOS PÚBLICOS

Concursos Públicos

Algoritmo de Ordenação por Inserção – Concurso BB (TI)

PUBLICADO POR

Antoniel da Silva Rego

TAGS:

agente de tecnologia / bando do brasil / TI

1 ANO ATRÁS

POSTS RELACIONADOS

TJ MT: Controle de Constitucionalidade

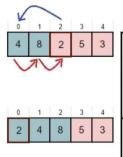


Concurso CAIXA: confira o cronograma após adiamento!



CAIXA: adiamento não prejudica isonomia, reafirma banco

ORDENAÇÃO POR INSERÇÃO



atual = 2;
Insira o elemento atual na posição correta entre os elementos que estão à suas esquerta (ordenados).

Comparações e movimentações:
8 > 2 ? SIM, então vetor[2] = 8;
4 > 2 ? SIM, então vetor[1] = 4;
vetor[0] = atual = 2;

O elemento inserido fará parte da lista ordenada:
- lista ordenada: [2, 4, 8];
- lista não ordenada: [5, 3];

Olá pessoal! Tudo bem? Para você que está se preparando para o concurso do Banco do Brasil, cargo Agente de Tecnologia, neste artigo estudaremos o algoritmo ordenação por inserção ou *insertion sort*.

Antes de mais nada, a ordenação de elementos consiste basicamente em colocá-los em **ordem crescente ou decrescente**. Alguns **exemplos** de ordenação são: ordenar alunos pelas suas notas; ordenar uma lista telefônica pelo nome das pessoas; ordenar pessoas por idade; ordenar clientes de acordo com a renda; entre outras.

Além disso, para ordenação dos elementos, utilizam-se diversos

