

# Insertion Sort

Murilo Dantas

# Insertion Sort

- Funcionamento

- ▶ As comparações iniciam a partir do segundo número do vetor de tamanho  $n$ .
- ▶ Números à esquerda do número escolhido, estão sempre ordenados.
- ▶ Um laço com as comparações será executado do segundo ao último elemento.

`for(i=1; i<n; i++)`

# Insertion Sort

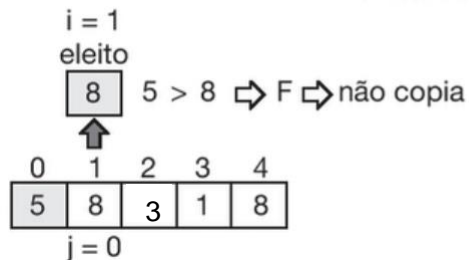
- Funcionamento (cont.)
  - ▶ Condições para executar o laço:
    - ▶ Enquanto existirem elementos à esquerda do número eleito para comparações.
    - ▶ A posição que atende a ordenação que se busca não for encontrada.

# Insertion Sort

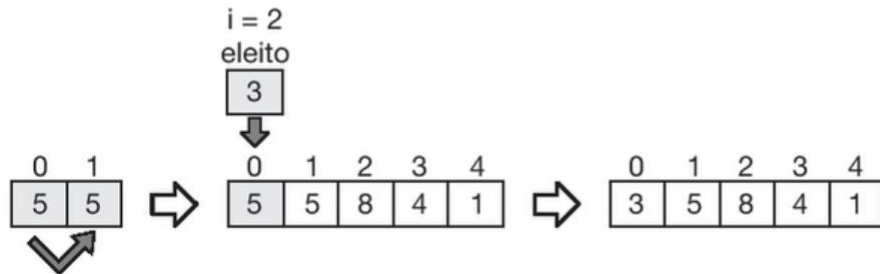
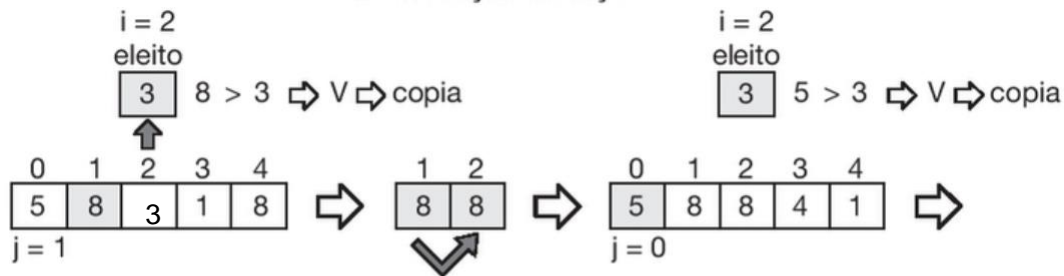
- Funcionamento (cont.)
  - ▶ O número eleito está na posição  $i$  e os números à esquerda estão na posição  $i-1$  a  $0$ .
  - ▶ O segundo laço será:  
 $j=i-1$  e `while(j >= 0 && elemento[j] > eleito)`

# Insertion Sort: exemplo

## 1ª execução do laço

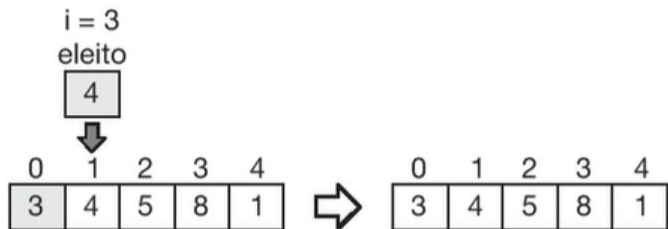
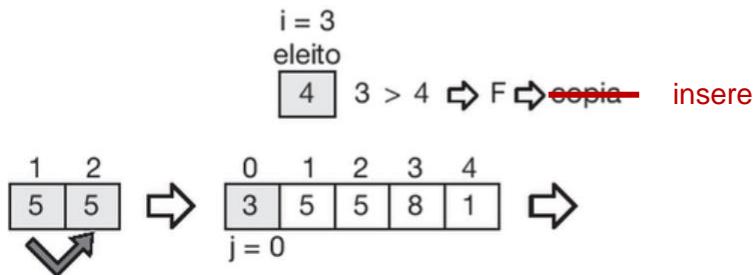
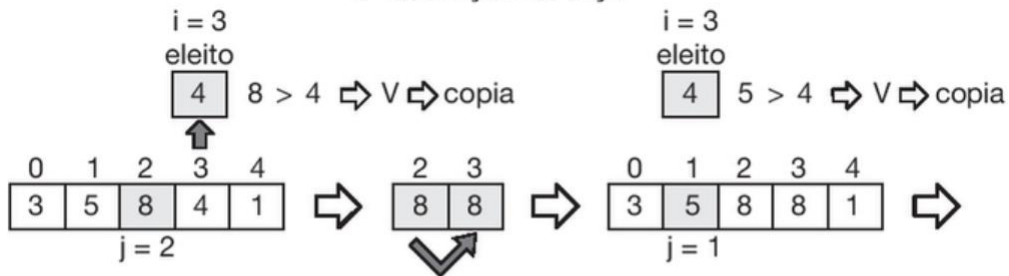


## 2ª execução do laço



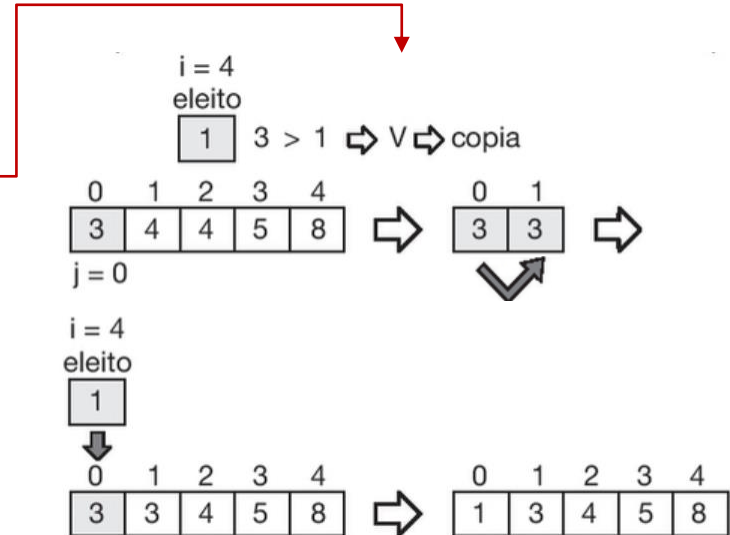
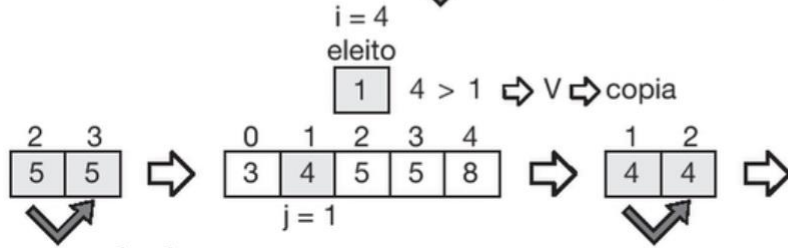
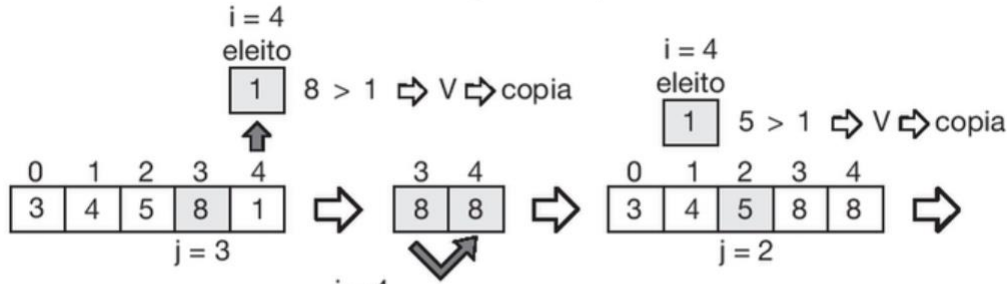
# Insertion Sort: exemplo

3ª execução do laço



# Insertion Sort: exemplo

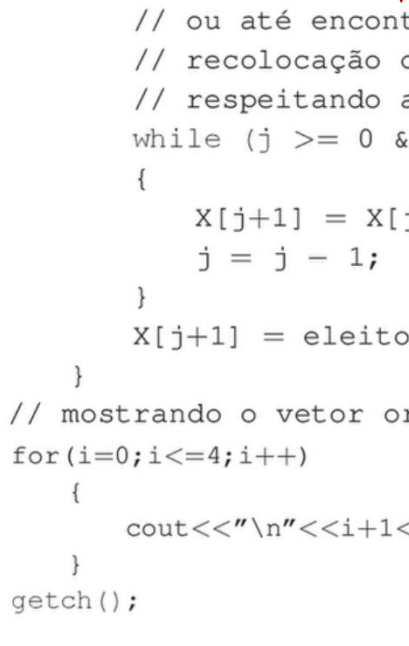
4ª execução do laço



# Insertion Sort: código

```
#include <iostream.h>
#include <conio.h>

void main()
{
    int X[5];
    int i, j, eleito;
    clrscr();
    // carregando os números no vetor
    for(i=0;i<=4;i++)
    {
        cout<<"Digite o "<<i+1<<"º número: ";
        cin>>X[i];
    }
    // ordenando de forma crescente
    // laço com a quantidade de elementos do vetor - 1
    for(i=1;i<=4;i++)
    {
        eleito = X[i];
        j = i - 1;
        // laço que percorre os elementos à
        // esquerda do número eleito
```



```
        // ou até encontrar a posição para
        // recolocação do número eleito
        // respeitando a ordenação procurada
        while (j >= 0 && X[j] > eleito)
        {
            X[j+1] = X[j];
            j = j - 1;
        }
        X[j+1] = eleito;
    }
    // mostrando o vetor ordenado
    for(i=0;i<=4;i++)
    {
        cout<<"\n"<<i+1<<"º número: "<<X[i];
    }
    getch();
}
```



**Perguntas?**

# Bibliografia da aula

- ASCENCIO, A. F. G.; ARAÚJO, G. S. Estrutura de dados. Algoritmos, análise da complexidade e implementação em Java e C/C++. Pearson. 2010.