

**UNINOVE**

# **Estrutura de Dados**

Prof. Mailson

Email: [mailson.silva@uninove.br](mailto:mailson.silva@uninove.br)



# Ordenação de Vetor Selection Sort

Este método consiste em selecionar o menor valor do vetor e movê-lo para o primeiro índice, a seguir, seleciona-se o próximo menor valor, desconsiderando o já colocado na primeira posição.

6	2	1	3	<u>4</u>	5	8	7	0
---	---	---	---	----------	---	---	---	---

- <http://www.youtube.com/watch?v=BSXlolkG5F8>

## Ordenação por seleção (Selection Sort)

```
void ssort ( int v[ ], int t )
{
    int i, j, min, k ;
    for ( i = 0; i < ( t - 1 ) ; i++ )
    {
        min = i ;
        for ( j = ( i + 1 ) ; j < t ; j++ )
        {
            if ( v[ j ] < v[ min ] )
                min = j;
        }
        if ( i != min )
        {
            k = v[ i ];
            v[ i ] = v[ min ];
            v[ min ] = k;
        }
    }
}
```

v – recebe o vetor que será ordenado;

t – recebe a quantidade de elementos do vetor;

i – determina o número de etapas para ordenação;

j – determina o número de comparações em cada etapa e os índices a serem pesquisados para a comparação;

k – variável auxiliar para ajudar na troca de posição dos valores no vetor;

min – armazena o índice que contém o menor valor do vetor e o índice a ser pesquisado para a comparação;

```
void ssort ( int v[ ], int t )
{
    int i, j, min, k ;
    for ( i = 0; i < ( t - 1 ) ; i++ )
    {
        min = i ;
        for ( j = ( i + 1 ) ; j < t ; j++ )
        {
            if ( v[ j ] < v[ min ] )
                min = j;
        }
        if ( i != min )
        {
            k = v[ i ];
            v[ i ] = v[ min ];
            v[ min ] = k;
        }
    }
}
```

[illegible]

# Ordenação de Vetor Insertion Sort

## Ordenação por inserção (Insertion Sort)

Este método consiste em inserir cada um dos elementos em sua posição correta relativa à sequência ordenada.

6	2	1	3	<u>4</u>	5	8	7	0
---	---	---	---	----------	---	---	---	---

- <https://www.youtube.com/watch?v=-Z00it6Nkz8>

[illegible]



## Ordenação por inserção (Insertion Sort)

```
void isort(int v[], int t)
{
    int i, j, k;
    for (i = 1; i < t; i++)
    {
        k = v[i];
        j = i-1;
        while (j >= 0 && v[j] > k)
        {
            v[j+1] = v[j];
            j--;
        }
        v[j+1] = k;
    }
}
```

v – recebe o vetor que será ordenado;

t – recebe a quantidade de elementos do vetor;

i – determina o número de etapas para ordenação;

j – determina o número de comparações em cada etapa e os índices a serem pesquisados para a comparação;

k – variável auxiliar para ajudar na troca de posição dos valores no vetor;

- 1 – Desenvolva um programa que contenha três vetores que devem ser preenchidos com 10 números cada um. Crie um procedimento que ordene o primeiro vetor utilizando o método de trocas, um procedimento que ordene o segundo vetor utilizando o método de seleção e um procedimento que ordene o terceiro vetor utilizando o método de inserção. Ao final mostre o conteúdo dos três vetores na tela;
  
- 2 – Refaça o exercício anterior mudando os vetores para que contenha 5 nomes cada um;