



FUNDAMENTOS DA CIÊNCIA DA COMPUTAÇÃO

Apresentação do Segundo Bimestre

“Métodos de ordenação: Selection Sort”

1. GRUPO

Nome : **Marina Cantarelli Barroca**

RA: **10740412**

Nome: **Katarina Angel de Souza Garcia**

RA: **10737248**

Nome: **Luiza Tafner Pereira**

RA: **10742953**



2. INTRODUÇÃO

O **Selection Sort** (ou ordenação por seleção) é um algoritmo de ordenação simples que organiza uma lista de elementos em ordem crescente ou decrescente. Ele é conhecido pela sua facilidade de implementação, embora não seja o mais eficiente para grandes conjuntos de dados.

3. EXPLICAÇÃO DO MÉTODO

O Selection Sort opera dividindo a lista em duas partes: uma sublista ordenada e uma sublista não ordenada. Inicialmente, a sublista ordenada está vazia, e a sublista não ordenada contém todos os elementos. O algoritmo segue as seguintes etapas:

1. **Encontrar o menor (ou maior) elemento:** Na sublista não ordenada, identifica-se o elemento de menor valor (para ordenação crescente) ou maior valor (para ordenação decrescente).
2. **Troca com o primeiro elemento da sublista não ordenada:** O elemento encontrado é trocado com o primeiro elemento da sublista não ordenada, movendo-o para a sublista ordenada.
3. **Reduzir a sublista não ordenada:** A sublista não ordenada é reduzida, avançando o início da sublista para o próximo elemento.
4. **Repetir:** O processo é repetido até que a sublista não ordenada esteja vazia, ou seja, até que todos os elementos estejam na sublista ordenada.

4. CARACTERÍSTICAS

- **Estabilidade:** O Selection Sort não é estável, ou seja, pode alterar a ordem relativa de elementos iguais.
- **Vantagens:**
 - Simplicidade de implementação.
 - Eficiente em termos de espaço (in-place).
 - Útil para listas pequenas ou quando o custo de escrita em memória é alto, pois minimiza trocas.
- **Desvantagens:**
 - Não é adaptável a listas parcialmente ordenadas.

5. APLICAÇÕES

O Selection Sort é usado em cenários onde:

- A lista a ser ordenada é pequena.
- A simplicidade do código é mais importante que a eficiência.



6. CÓDIGO EM PYTHON E O PSEUDOCÓDIGO

Algoritmo SelectionSort

Entrada: Uma lista de números chamada lista

Saída: A lista ordenada em ordem crescente

Função selection_sort(lista)

tamanho ← tamanho da lista

Para i de 0 até tamanho - 1 faça

menor ← i // Supõe que o menor elemento está na posição i

Para j de i + 1 até tamanho - 1 faça

Se lista[j] < lista[menor] então

menor ← j // Atualiza a posição do menor elemento encontrado

FimSe

FimPara

// Troca os elementos das posições i e menor

temp ← lista[i]

lista[i] ← lista[menor]

lista[menor] ← temp



FimPara

FimFunção

// Programa principal

Início

$n \leftarrow [2, 150, 3, 0, 9, 48, 8, 35, 5]$

Chamar selection_sort(n)

Escreva "Lista ordenada: ", n

Fim

```
def selection_sort(lista): # Função que implementa o selection sort
    com a intenção de ordenar a lista

    tamanho = len(lista) # Obtém o tamanho da lista pra ser usado
    nos loops

    # Percorre cada posição da lista

    for i in range(tamanho):

        menor = i # Vamos supor que o menor número está na posição
        i

        # Procurando o menor número no restante da lista

        for j in range(i + 1, tamanho):

            if lista[j] < lista[menor]: # Compara o elemento na
                posição j com o elemento na posição 'menor'
```



```
        menor = j # Achamos um número menor, guardamos a
        posição dele

    # Trocando os números de lugar

    lista[i], lista[menor] = lista[menor], lista[i]

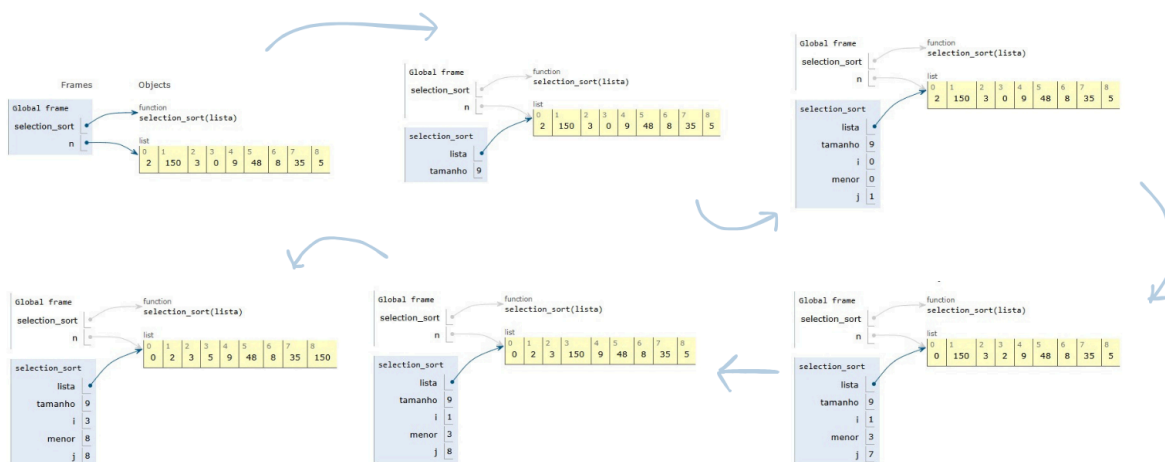
# main

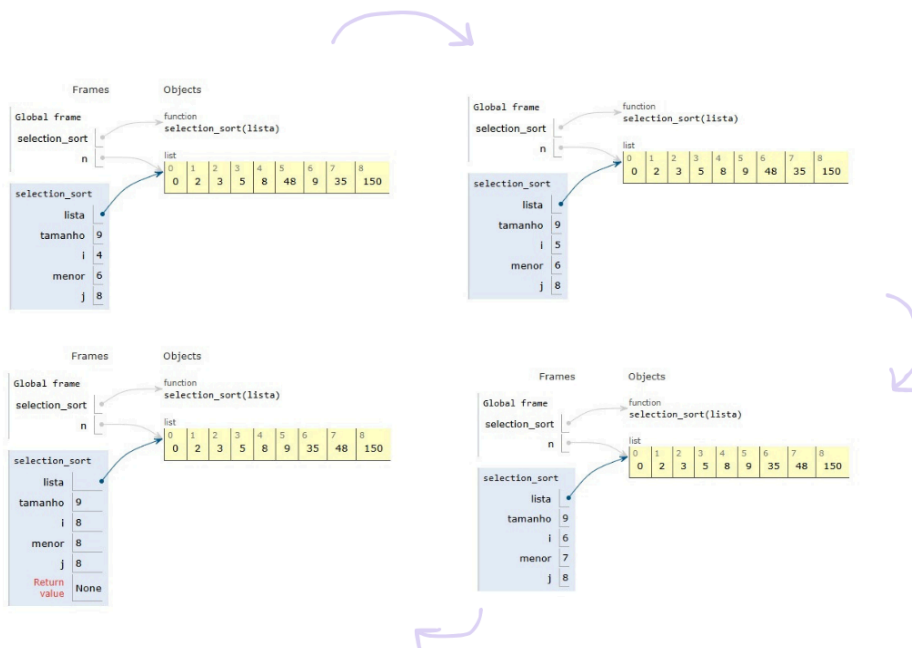
n = [2, 150, 3, 0, 9, 48, 8, 35, 5] # Cria uma lista desordenada
para teste

selection_sort(n) # Chama a função selection_sort para ordenar a
lista n

print("Lista ordenada: " , n) # Vai mostrar [0, 2, 3, 5, 8, 9, 35,
48, 150]
```

7. TESTE





8. CONCLUSÃO

O Selection Sort é um algoritmo de ordenação intuitivo e fácil de implementar, ideal para listas pequenas ou casos onde a simplicidade é prioritária. Apesar de sua complexidade limitar seu uso em grandes conjuntos de dados, ele é uma ferramenta valiosa em contextos específicos, como sistemas com restrições de memória ou aplicações educacionais para ensinar conceitos de ordenação.

Link do vídeo demonstrando: https://youtu.be/xxl2eD2Vu70?si=Kc_Ni77pmyXaucJ6

9. REFERÊNCIAS

AWARI. Seleção de ordenação em Python: aprenda a implementar o selection sort. 24 nov. 2023. Disponível em: <https://awari.com.br/selecao-de-ordenacao-em-python-aprenda-a-implementar-o-selection-sort/>. Acesso em: 25 maio 2025.

COELHO, Hebert; FÉLIX, Nádia. *Métodos de ordenação: Selection, Insertion, Bubble, Merge (Sort)*. [S. l.]: Universidade Federal de Goiás, Instituto de Informática. Disponível em: https://ww2.inf.ufg.br/~hebert/disc/aed1/AED1_04_ordenacao1.pdf. Acesso em: 25 maio 2025.