

# Pesquisa Métodos de Ordenação:

**Aluno: Guilherme Augusto dos Santos**

**RA: 24027387-2**

**Turma: ESOFT1S – E**

## SELECTION SHORT:

### Como funciona:

- Encontra o menor elemento no array e troca com o primeiro.
- Repetido para os elementos restantes, posicionando cada um na sua posição final.

A ordenação por seleção é um algoritmo baseado em passar sempre o menor valor do vetor para a primeira posição, depois o de segundo menor valor para a segunda posição e assim é feito sucessivamente com os  $(n-1)$  elementos restantes, até os últimos dois elementos

o numero de comparações é conhecido aplicando a fórmula  $TC = n*(n-1)/2$

é um método simples mas não é muito eficiente

### Vantagens do Selection Sort:

- **Simplicidade:** Fácil de entender e implementar, ideal para aprendizado inicial de algoritmos de ordenação.

### Desvantagens do Selection Sort:

- **Não é estável:**  
Pode alterar a ordem relativa de elementos iguais, o que é problemático em algumas aplicações.

```

void selectionSort(int arr[], int n) {
    for (int i = 0; i < n - 1; i++) {
        int minIndex = i;
        for (int j = i + 1; j < n; j++) {
            if (arr[j] < arr[minIndex]) {
                minIndex = j;
            }
        }
        int temp = arr[i];
        arr[i] = arr[minIndex];
        arr[minIndex] = temp;
    }
}

```

## BUBBLE SHORT:

### Como funciona:

- Compara elementos adjacentes no array.
- Troca se o atual for maior que o próximo.
- Após cada iteração, o maior elemento é colocado na posição correta, como uma bolha que sobe.

Conhecido como Algoritmo da bolha, tem por objetivo comparar elementos dois a dois empurrando

o maior elemento para o fim da tabela.

uma tabela com 5 elementos:

passo1: 1,2 2,3 3,4 4,5

passo2: 1,2 2,3 3,4

passo3: 1,2 2,3

passo4: 1,2

o numero de comparações é conhecido aplicando a formula  $TC = n \cdot (n-1) / 2$

## Vantagens e Desvantagens do Bubble Sort:

- **Vantagens:** Simples de implementar, estável (não muda a ordem de elementos iguais).
- **Desvantagens:** Ineficiente para grandes conjuntos de dados ( $O(n^2)$ ).

```
void bubbleSort(int arr[], int n) {  
    for (int i = 0; i < n - 1; i++) {  
        for (int j = 0; j < n - i - 1; j++) {  
            if (arr[j] > arr[j + 1]) {  
                int temp = arr[j];  
                arr[j] = arr[j + 1];  
                arr[j + 1] = temp;  
            }  
        }  
    }  
}
```

## INSERTION SHORT:

### Como funciona:

- Começa com o segundo elemento e o insere no lugar correto em relação aos elementos anteriores.
- Repetido para todos os elementos, ordenando progressivamente.

simples e eficiente quando aplicado em pequenas listas. A lista é percorrida do segundo até o ultimo elemento, comparando com o elemento anterior e trocando de lugar quando encontrado um valor menor (objetivo crescente). Funciona da mesma forma que as pessoas usam para ordenar cartas em um jogo de baralho como poker

o numero de comparações é mais dinamico, ou seja, não é possível determinar um formula. Em uma tabela contando 20 elementos, o pior caso será 190 comparações igual aos outros, mas no melhor caso será igual a 0 comparações

## Vantagens e Desvantagens do Bubble Sort:

- **Vantagens:** Simples, estável, eficiente para pequenos conjuntos ou listas quase ordenadas.
- **Desvantagens:** Desempenho cai com listas desordenadas e grandes ( $O(n^2)$ ).

```
void insertionSort(int arr[], int n) {  
    for (int i = 1; i < n; i++) {  
        int key = arr[i];  
        int j = i - 1;  
        while (j >= 0 && arr[j] > key) {  
            arr[j + 1] = arr[j];  
            j--;  
        }  
        arr[j + 1] = key;  
    }  
}
```