

ESTRUTURA DE DADOS

Ordenação de Listas (sort)



Roteiro

- Introdução
- Principais tipos de ordenação
 - ☐ **Insertion Sort**
 - ☐ **Bubble Sort**
 - ☐ **Selection Sort**
 - ☐ Merge Sort
 - ☐ Quick sort
 - ☐ Heap Sort
- Complexidade algorítmica

Introdução

Ordenação (ou sort) é um dos problemas fundamentais em ciência da computação e é amplamente utilizado em diversas áreas, como:

- processamento de dados,
- banco de dados,
- análise de dados,
- entre outros.

Introdução

Tipos de anúncio

Todos os anúncios



Ordenar por

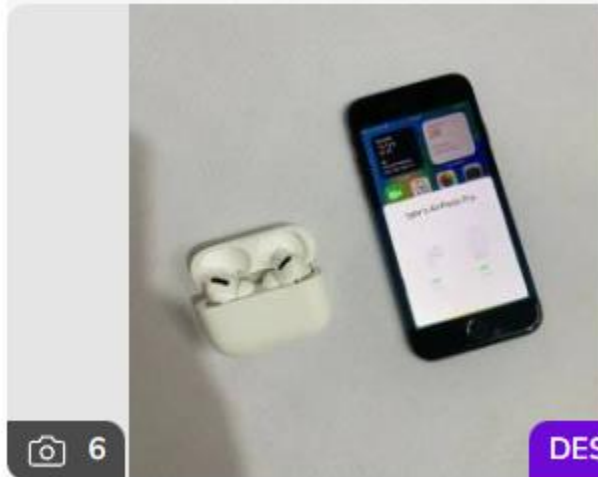
Mais Recentes

Mais Recentes

Mais Relevantes

Menor Preço

Maior Preço



6

DESTAQUE

Iphone 8 256gb + Airpod Pro Tudo i
caixa 100% Funcionando

ONLINE



Compra Segura



Parcelamento sem juros



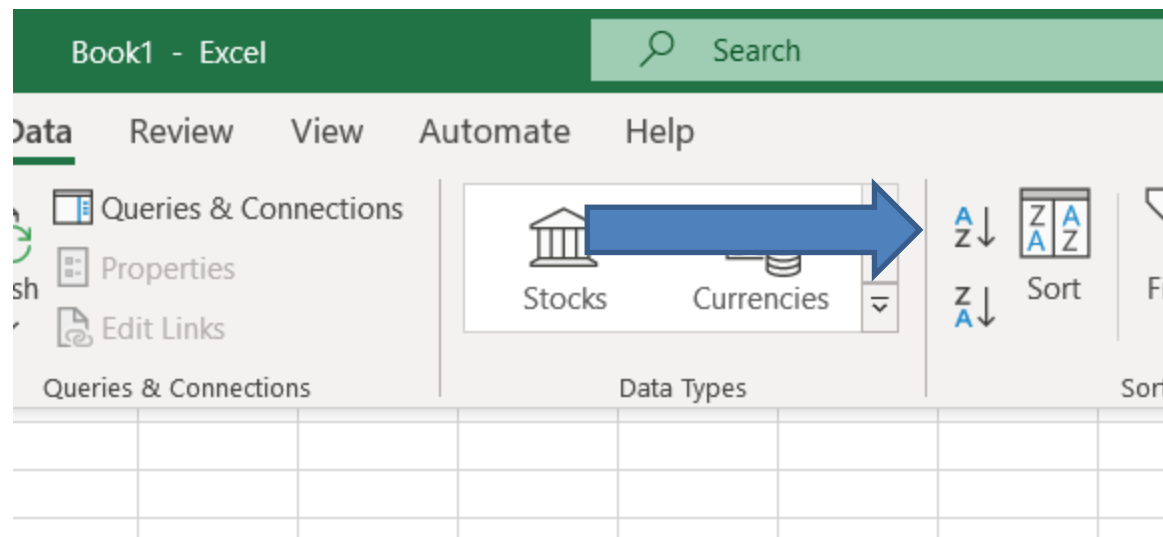
Desconto no frete



Delivery

São Caetano do Sul, São José

Introdução



Introdução

Existem vários tipos de algoritmos de ordenação, cada um com sua própria complexidade e eficiência, podemos citar como “principais”:

- Insertion Sort
- Bubble Sort
- Selection Sort
- Merge Sort
- Quick sort
- Heap Sort

Dando um chute

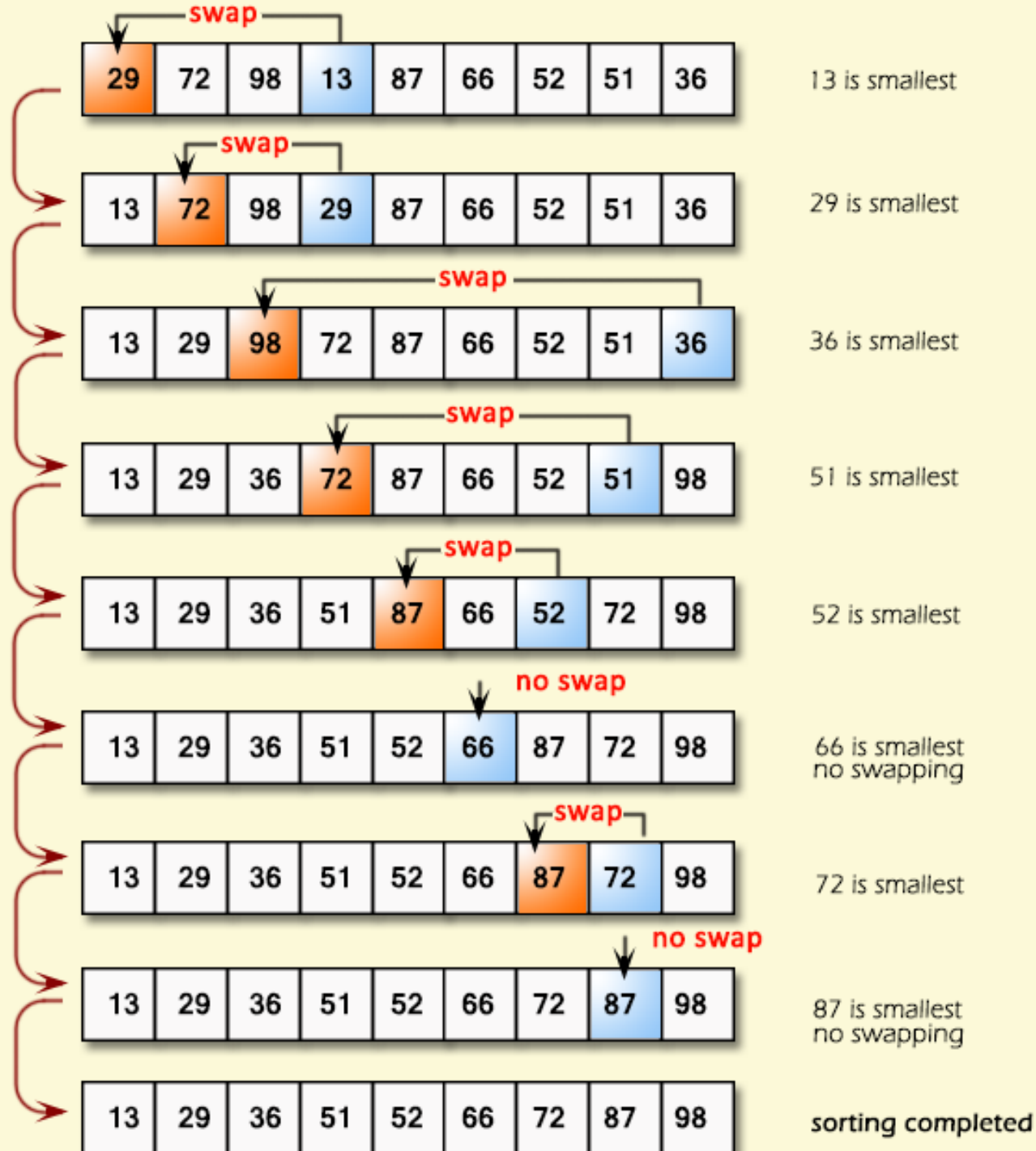
Vamos pensar em um algoritmo que ordene um vetor de números.

Exemplo:

Antes: [7, 9, 5, 6, 7, 2]

Depois: [2, 5, 6, 7, 7, 9]

Selection Sort



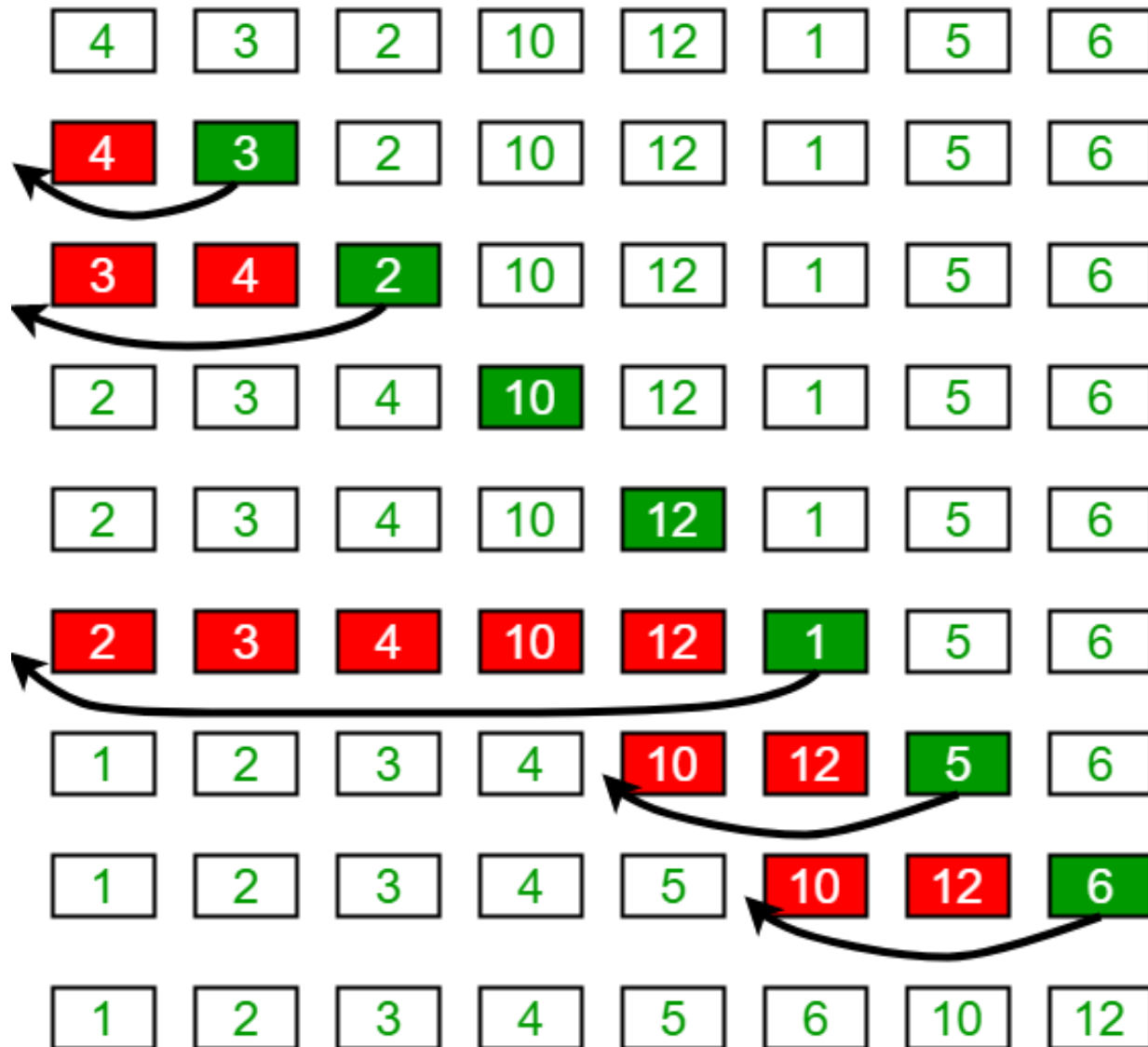
Selection Sort

```
public static void selectionSort(int[] arr) {  
    int n = arr.length;  
  
    // percorre o array  
    for (int i = 0; i < n - 1; i++) {  
        // encontra o menor elemento no array desordenado  
        int minIndex = i;  
        for (int j = i + 1; j < n; j++) {  
            if (arr[j] < arr[minIndex]) {  
                minIndex = j;  
            }  
        }  
  
        // troca o menor elemento com o primeiro elemento do array desordenado  
        int temp = arr[minIndex];  
        arr[minIndex] = arr[i];  
        arr[i] = temp;  
    }  
}
```

Selection Sort

Selection Sort	$\Omega(N^2)$	$\Theta(N^2)$	$O(N^2)$	$O(1)$
-----------------------	---------------	---------------	----------	--------

Insertion Sort



Insertion Sort

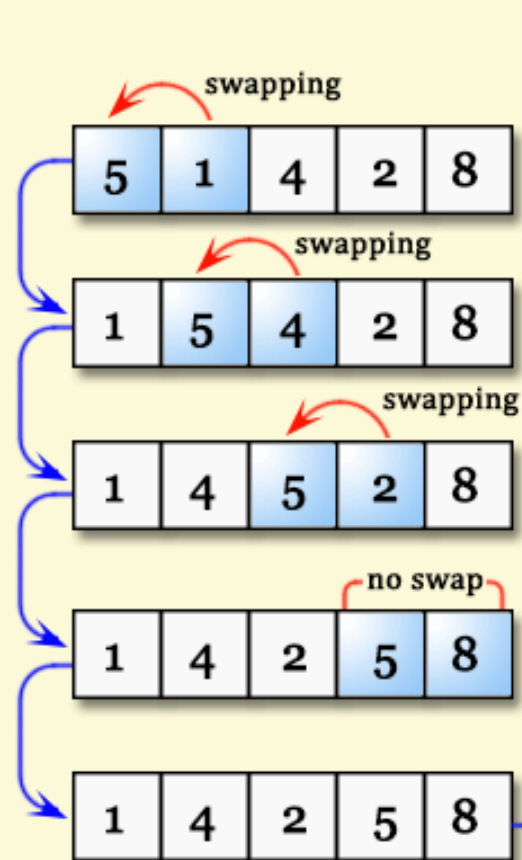
```
public static void insertionSort(int[] arr) {  
    int n = arr.length;  
    for (int i = 1; i < n; i++) {  
        int key = arr[i];  
        int j = i - 1;  
        while (j >= 0 && arr[j] > key) {  
            arr[j+1] = arr[j];  
            j = j - 1;  
        }  
        arr[j+1] = key;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

Insertion Sort

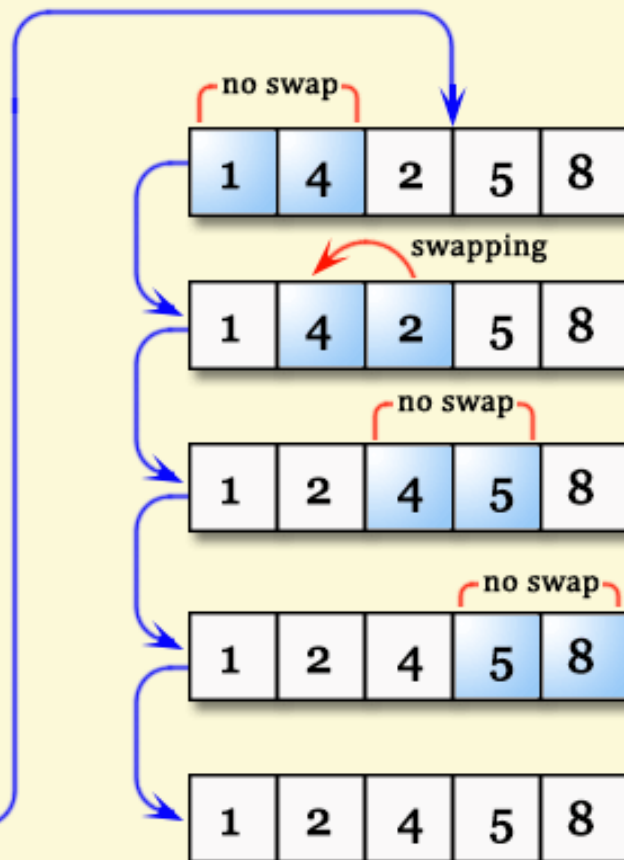
Insertion Sort	$\Omega(N)$	$\Theta(N^2)$	$O(N^2)$
----------------	-------------	---------------	----------

Bubble Sort

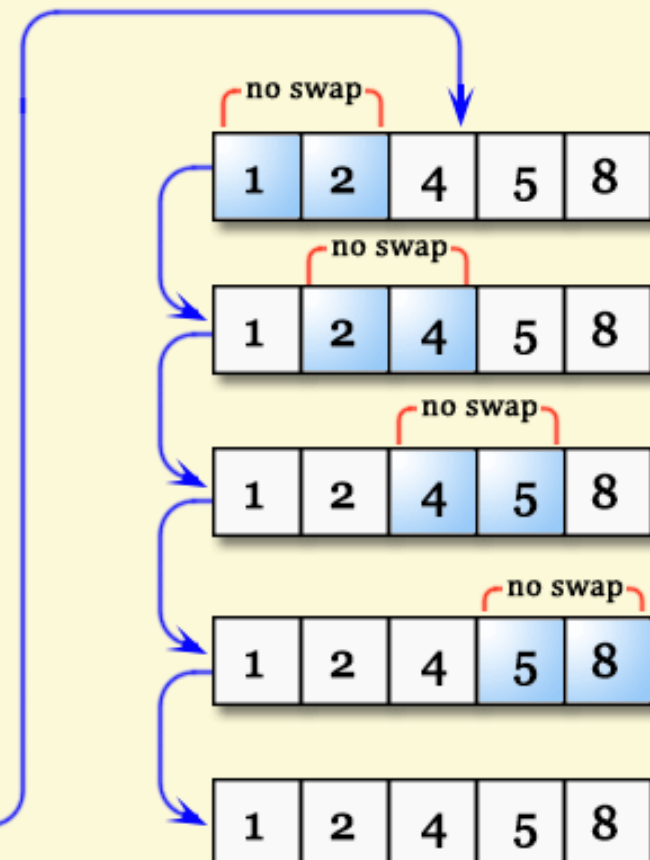
First Pass



Second Pass



Third Pass



Bubble Sort

```
public static void bubbleSort(int[] arr) {  
    int n = arr.length;  
    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;  
            }  
        }  
    }  
}
```


Bubble Sort

```
public static void bubbleSort(int[] arr) {
    int n = arr.length;
    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;
            }
        }
    }
}
```

Bubble Sort	$\Omega(N)$	$\Theta(N^2)$	$O(N^2)$	$O(1)$
-------------	-------------	---------------	----------	--------

```
}
```

Bubble Sort

Como tornar o Bubble sort $\Omega(n)$?

Complexidade algorítmica

Sorting Algorithms	Time Complexity		
	Best Case	Average Case	Worst Case
Bubble Sort	$\Omega(N)$	$\Theta(N^2)$	$O(N^2)$
Selection Sort	$\Omega(N^2)$	$\Theta(N^2)$	$O(N^2)$
Insertion Sort	$\Omega(N)$	$\Theta(N^2)$	$O(N^2)$
Quick Sort	$\Omega(N \log N)$	$\Theta(N \log N)$	$O(N^2)$
Merge Sort	$\Omega(N \log N)$	$\Theta(N \log N)$	$O(N \log N)$
Heap Sort	$\Omega(N \log N)$	$\Theta(N \log N)$	$O(N \log N)$

Spoiler próxima aula:

Sorting Algorithms	Time Complexity		
	Best Case	Average Case	Worst Case
Bubble Sort	$\Omega(N)$	$\Theta(N^2)$	$O(N^2)$
Selection Sort	$\Omega(N^2)$	$\Theta(N^2)$	$O(N^2)$
Insertion Sort	$\Omega(N)$	$\Theta(N^2)$	$O(N^2)$
Quick Sort	$\Omega(N \log N)$	$\Theta(N \log N)$	$O(N^2)$
Merge Sort	$\Omega(N \log N)$	$\Theta(N \log N)$	$O(N \log N)$
Heap Sort	$\Omega(N \log N)$	$\Theta(N \log N)$	$O(N \log N)$

Exercícios

Dado o seguinte vetor:

[4, 6, 1, 7, 4, 8]

Mostre como os valores vão ficando a **cada iteração** de cada um dos algoritmos:

A – Selection sort

B – Insertion sort

C – Bubble sort