

TUTORIA 06

MÉTODOS DE ORDENAÇÃO

Questão 01.

a) [10, 82, 91, 42, 1, 9, 30, 31]

Bubble Sort:

[10, 82, 42, 1, 9, 30, 31, 91]

[10, 42, 1, 9, 30, 31, 82, 91]

[10, 1, 9, 30, 31, 42, 82, 91]

[1, 9, 10, 30, 31, 42, 82, 91]

Resultado:

[1, 9, 10, 30, 31, 42, 82, 91]

Insertion Sort:

[10, 82, 91, 42, 1, 9, 30, 31]

[10, 82, 91, 42, 1, 9, 30, 31]

[10, 42, 82, 91, 1, 9, 30, 31]

[1, 10, 42, 82, 91, 9, 30, 31]

[1, 9, 10, 42, 82, 91, 30, 31]

[1, 9, 10, 30, 42, 82, 91, 31]

[1, 9, 10, 30, 31, 42, 82, 91]

Resultado:

[1, 9, 10, 30, 31, 42, 82, 91]

Selection Sort:

[1, 82, 91, 42, 10, 9, 30, 31]

[1, 9, 91, 42, 10, 82, 30, 31]

[1, 9, 10, 42, 91, 82, 30, 31]

[1, 9, 10, 30, 91, 82, 42, 31]

[1, 9, 10, 30, 31, 82, 42, 91]

[1, 9, 10, 30, 31, 42, 82, 91]

Resultado:

[1, 9, 10, 30, 31, 42, 82, 91]

b) [90, 85, 80, 75, 70, 65, 60, 55]

Bubble Sort:

[85, 80, 75, 70, 65, 60, 55, 90]
[80, 75, 70, 65, 60, 55, 85, 90]
[75, 70, 65, 60, 55, 80, 85, 90]
[70, 65, 60, 55, 75, 80, 85, 90]
[65, 60, 55, 70, 75, 80, 85, 90]
[60, 55, 65, 70, 75, 80, 85, 90]
[55, 60, 65, 70, 75, 80, 85, 90]

Resultado:

[55, 60, 65, 70, 75, 80, 85, 90]

Insertion Sort:

[85, 90, 80, 75, 70, 65, 60, 55]
[80, 85, 90, 75, 70, 65, 60, 55]
[75, 80, 85, 90, 70, 65, 60, 55]
[70, 75, 80, 85, 90, 65, 60, 55]
[65, 70, 75, 80, 85, 90, 60, 55]
[60, 65, 70, 75, 80, 85, 90, 55]
[55, 60, 65, 70, 75, 80, 85, 90]

Resultado:

[55, 60, 65, 70, 75, 80, 85, 90]

Selection Sort:

[55, 85, 80, 75, 70, 65, 60, 90]
[55, 60, 80, 75, 70, 65, 85, 90]
[55, 60, 65, 75, 70, 80, 85, 90]
[55, 60, 65, 70, 75, 80, 85, 90]
[55, 60, 65, 70, 75, 80, 85, 90]
[55, 60, 65, 70, 75, 80, 85, 90]

Resultado:

[55, 60, 65, 70, 75, 80, 85, 90]

c) [3, 5, 7, 9, 11, 13, 15, 1]

Bubble Sort:

[3, 5, 7, 9, 11, 13, 1, 15]
[3, 5, 7, 9, 11, 1, 13, 15]
[3, 5, 7, 9, 1, 11, 13, 15]
[3, 5, 7, 1, 9, 11, 13, 15]
[3, 5, 1, 7, 9, 11, 13, 15]
[3, 1, 5, 7, 9, 11, 13, 15]
[1, 3, 5, 7, 9, 11, 13, 15]

Resultado:

[1, 3, 5, 7, 9, 11, 13, 15]

Insertion Sort:

[3, 5, 7, 9, 11, 13, 15, 1]
[3, 5, 7, 9, 11, 13, 15, 1]
[3, 5, 7, 9, 11, 13, 15, 1]
[3, 5, 7, 9, 11, 13, 15, 1]
[3, 5, 7, 9, 11, 13, 15, 1]
[3, 5, 7, 9, 11, 13, 15, 1]
[3, 5, 7, 9, 11, 13, 15, 1]
[1, 3, 5, 7, 9, 11, 13, 15]

Resultado:

[1, 3, 5, 7, 9, 11, 13, 15]

Selection Sort:

[1, 5, 7, 9, 11, 13, 15, 3]
[1, 3, 7, 9, 11, 13, 15, 5]
[1, 3, 5, 9, 11, 13, 15, 7]
[1, 3, 5, 7, 11, 13, 15, 9]
[1, 3, 5, 7, 9, 13, 15, 11]
[1, 3, 5, 7, 9, 11, 15, 13]
[1, 3, 5, 7, 9, 11, 13, 15]

Resultado:

[1, 3, 5, 7, 9, 11, 13, 15]

Questão 02.

Um algoritmo de ordenação é estável quando os elementos com valores iguais mantêm a mesma ordem relativa após a ordenação. Ou seja, se mantém na mesma ordem que estavam antes da ordenação ser realizada.

Um algoritmo é in situ quando realiza a ordenação usando apenas uma pequena quantidade extra de memória, normalmente O(1). Ou seja, ele não cria vetores adicionais, faz tudo dentro do próprio arranjo original.

1. Bubble Sort - Estável e In situ

Troca apenas quando o elemento da esquerda é maior, não troca elementos iguais. Utiliza apenas variáveis auxiliares simples.

2. Insertion Sort - Estável e In situ

Ao inserir um elemento, elementos iguais não são ultrapassados. Só utiliza variáveis auxiliares (chave para inserção).

3) Selection Sort - In situ

Troca sempre o menor elemento com o atual, isso pode inverter a ordem relativa de elementos iguais. Só faz trocas dentro do próprio vetor.

Questão 03.

[8, 2, 6, 4, 5, 1, 3, 7] → [8, 2, 6, 4] | [5, 1, 3, 7]

[8, 2, 6, 4] → [8, 2] | [6, 4]

[8, 2] → [8] | [2]

[6, 4] → [6] | [4]

[5, 1, 3, 7] → [5, 1] | [3, 7]

[5, 1] → [5] | [1]

[3, 7] → [3] | [7]

[8] + [2] → [2, 8]

[6] + [4] → [4, 6]

[2, 8] + [4, 6] → [2, 4, 6, 8]

[5] + [1] → [1, 5]

[3] + [7] → [3, 7]

[1, 5] + [3, 7] > [1, 3, 5, 7]

[2, 4, 6, 8] + [1, 3, 5, 7] → [1, 2, 3, 4, 5, 6, 7, 8]

Resultado:

[1, 2, 3, 4, 5, 6, 7, 8]

Questão 04.

Algoritmo	Comparações	Movimentações	Espaço Adicional
Bubble Sort	$O(n^2)$	$O(n^2)$	$O(1)$
Insertion Sort	$O(n^2)$	$O(n^2)$	$O(1)$
Selection Sort	$O(n^2)$	$O(n)$	$O(1)$
Merge Sort	$O(n \log n)$	$O(n \log n)$	$O(n)$