

Dynamic Programming

Lista de Exercícios

Gabriel Sobral

29 de agosto de 2024

Questão 1.

Modifique os algoritmos *InsertionSort*, *SelectionSort* e *BubbleSort* para que eles ordem um vetor em ordem *não crescente*. Por exemplo, dado um vetor com os elementos $[5, 3, 3, 19, 8]$ o algoritmo deve retornar $[19, 8, 5, 3, 3]$.

Questão 2.

Dado um vetor V com os seguintes elementos $V = [31, 41, 59, 26, 41, 58]$. Execute os algoritmos *InsertionSort*, *SelectionSort*, *BubbleSort* e conte o número de trocas. Faça uma análise do número de trocas desses algoritmos.

Questão 3.

Considere um vetor V ordenado em ordem crescente de n elementos, onde $n > 1$ (V contém dois elementos ou mais). Os algoritmos *InsertionSort* e *SelectionSort* efetuam o mesmo número de trocas? Você pode usar um exemplo para explicar o seu raciocínio.

Questão 4.

Considere um vetor V ordenado em ordem **decrescente** de n elementos, onde $n > 1$ (V contém dois elementos ou mais). Os algoritmos *BubbleSort* e *SelectionSort* efetuam o mesmo número de trocas? Você pode usar um exemplo para explicar o seu raciocínio.

Questão 5.

Execute o *Combina*, do algoritmo *MergeSort*, para combinar os vetores $E = [26, 58, 59]$ e $D = [31, 41]$.

Questão 6.

Execute o *MergeSort* para ordenar o vetor $V = [41, 26, 58, 59, 31]$.

Questão 7.

Execute o *Particiona*, do algoritmo *QuickSort*, nos vetores $V_1 = [41, 26, 58, 59, 31]$, $V_2 = [26, 31, 41, 58, 59]$ e anote a posição do pivot que resultar da execução do *Particiona*. O pivot particiona os vetores V_1 e V_2 da mesma forma?

Questão 8.

Execute o *QuickSort* para ordenar o vetor $V = [41, 26, 58, 59, 31]$.