

5ª Lista de Exercícios

BUSCA E ORDENAÇÃO

1) Faça um procedimento em C para a busca sequencial em listas encadeadas usando recursividade.

2) Assim como a busca binária, a busca ternária é um algoritmo de busca que encontra a posição do valor alvo (valor a ser pesquisado) dentro de um array ordenado. A busca binária é usada para dividir o array em duas partes, mas a busca ternária, como o nome sugere, subdivide o array em três partes. O algoritmo funciona como a seguir. Considere um array ordenado (ordem crescente ou decrescente) *arr* de *n* elementos, e um elemento *x* deve ser pesquisado, quer esteja presente no array ou não. A estratégia é dividir repetidamente o array em três partes iguais. Portanto, deve haver dois valores médios, digamos *med1* e *med2*, e subarrays seriam como, um $[0, 1, \dots, (med1-1)]$, um $[(med1 + 1), \dots, (med2-1)]$ e um $[(med2 + 1) \dots (ultimo)]$. Por favor, note, não é errado que o primeiro subarray não inclua *med1*, pois será melhor entendido seguindo as regras a seguir, onde pode ser observado que *med1* é verificado de antemão. As seguintes regras são seguidas repetidamente:

Se $x = med1$, significa que o número foi encontrado e o índice de *med1* é retornado.

Se $x = med2$, significa que o número foi encontrado e o índice de *med2* é retornado.

Se $x < med1$, pesquise a primeira parte do array.

Se $x > med2$, pesquise a terceira parte do array.

Senão, pesquise a segunda parte do array.

De acordo com as regras, implemente uma subrotina que implemente a busca ternária de forma recursiva, a qual deve retornar o índice do elemento encontrado (ou não) no array.

3) Dada a sequência: 19, 44, 01, 03, 23, 54, 25, 60, 04, represente a ordenação desta sequência de chaves na alternativa de ordenação:

- a) de **contiguidade física**;
- b) por **vetor indireto de ordenação**;
- c) por **encadeamento**.

4) Ilustre a operação de alguns dos algoritmos de ordenação que estudamos sobre o vetor $A = (3, 41, 52, 26, 38, 57, 9, 49)$.

5) Considerando o seguinte vetor: 9, 25, 10, 18, 5, 7, 15, 3, descreva todos os passos utilizados para a ordenação do mesmo utilizando o método *heap sort*.

6) A ordenação por inserção pode ser expressa como um procedimento recursivo da seguinte forma. Para ordenar o vetor $A[1..n]$, ordenamos recursivamente o vetor $A[1..n - 1]$ e então inserimos $A[n]$ no vetor ordenado $A[1..n - 1]$. Escreva um algoritmo de ordenação por inserção recursivo que utiliza essa ideia.

7) Implemente a versão recursiva do algoritmo de ordenação *BubbleSort*.

8) Implemente a versão recursiva do algoritmo de ordenação *InsertionSort*.

9) Imagine que você queira ordenar uma lista de pessoas de acordo com a idade de cada uma delas. Os nomes das pessoas estão em um vetor P , e outro vetor I contém a lista com as idades respectivas das pessoas, de tal modo que $I[x]$ é a idade da pessoa $P[x]$. Como podemos ordenar $I[x]$ sem perder a referência de qual idade pertence a qual pessoa? Faça um programa que resolva o problema, usando uma adaptação de qualquer dos algoritmos de ordenação apresentados.