

## AED2 - Lista 1

### Ordenações elementares + heap

Seguem alguns exercícios relacionados com algoritmos de ordenação.

1 - [8.1.1] - Escreva uma função que decida se um vetor  $v[0..n-1]$  está em ordem crescente. Depois, critique o código a seguir.

```
int verifica (int v[], int n) {  
    int i, anterior = v[0], sim = 1;  
    for (i = 1; i < n && sim; i++) {  
        if (anterior > v[i]) sim = 0;  
        anterior = v[i];  
    }  
    return sim;  
}
```

2 - [8.2.8, 8.3.3] - Modifique as seguintes funções de ordenação para que elas permutem os elementos de um vetor  $v[0..n-1]$  de modo que eles fiquem em ordem decrescente.

- a) insertionSort
- b) selectionSort
- c) bubbleSort

3 - [10.5.1] - [Ordem Decrescente] Escreva uma versão do algoritmo heapSort que rearranja um vetor  $v[0 .. n - 1]$  de modo que ele fique em ordem decrescente.

4 - [10.5.2] - [Ordenação de Strings] Escreva uma versão do algoritmo heapSort que coloque um vetor de strings em ordem lexicográfica (“ordem alfabética”). Veja <https://www.ime.usp.br/~pf/algoritmos/aulas/string.html>.

5 - [10.1.1] - Mostre que todo vetor decrescente é um max-heap. Mostre que a recíproca não é verdadeira.

6 - [10.1.3] - Escreva uma função que decida se um vetor  $v[0 .. m - 1]$  é ou não um max-heap.

Para revisar conceitos sobre ordenação, além de encontrar mais exercícios, acesse:

- <https://www.ime.usp.br/~pf/algoritmos/aulas/ordena.html>
- <https://www.ime.usp.br/~pf/algoritmos/aulas/hpsrt.html>