

**Lista 14****Ordenação (parte 2)****Questão 1**

Escreva uma função `intercala()` que recebe dois vetores **ordenados** A, de tamanho n, e B, de tamanho m, e retorna os valores de A e B em ordem em um terceiro vetor C. Sua função deverá possuir a seguinte assinatura:

```
1 int* intercala(int* A, int* B, int n, int m);
```

- (a) Faça uma análise teórica de pior e melhor caso do seu algoritmo.

**Questão 2**

Um algoritmo de intercalação é estável se não altera a posição relativa dos elementos que têm um mesmo valor. Por exemplo, se o vetor tiver dois elementos de valor 222, um algoritmo de intercalação estável manterá o primeiro 222 antes do segundo.

A função `intercala()` do **exercício anterior** é estável? Se não, é possível modificá-la para que seja?

**Questão 3**

Simule detalhadamente a execução do algoritmo `Merge Sort` sobre o vetor de entrada:

```
1 int v[] = {3, 41, 52, 26, 38, 57, 9, 49};
```

**Questão 4**

Escreva uma versão não recursiva do algoritmo `Merge Sort`.

**Questão 5**

Seja `A[0..n-1]` um vetor de  $n$  números inteiros distintos dois a dois.

Se  $i < j$  e  $A[i] > A[j]$  então o par  $(i, j)$  é chamado uma inversão de `A`.

- (a) Liste as cinco inversões do vetor `A[] = {2, 3, 8, 6, 1}`.
- (b) Qual vetor com elementos no conjunto  $\{1, 2, \dots, n\}$  tem a maior quantidade de inversões? Quantas são?
- (c) Escreva um programa que determine o número de inversões em qualquer permutação de  $n$  elementos em tempo de execução de pior caso  $O(n \log n)$ .

---

### Questão 6

Escreva uma função `particiona()` que recebe um vetor `A`, de tamanho `n`, e um pivo `p`, e retorna o vetor `A` com os valores menores do que `A[p]` à esquerda de `p` e os maiores a direita. Sua função deverá possuir a seguinte assinatura:

```
1 int particiona(int* A, int p);
```

(a) Faça uma análise teórica de pior e melhor caso do seu algoritmo.

### Questão 7

Ilustre a operação da função `particiona` do algoritmo `Quick Sort` sobre o vetor `v` a seguir.

```
1 int v[] = {3, 41, 52, 26, 38, 57, 9, 49};
```

### Questão 8

Descreva com suas palavras (alto nível) o algoritmo `Quick Sort`.

### Questão 9

Por quê no pior caso, o custo computacional do `Quick Sort` é  $O(n^2)$ ? Dê um exemplo em que o pior caso acontece.

### Questão 10

Escreva uma versão não recursiva do algoritmo `Quick Sort`.