

Aula 12 - Exercícios: Algoritmos de ordenação (parte 2)

1. Escreva uma função recursiva com cabeçalho

```
int maximo (int v[], int ini, int fim);
```

que devolve o maior valor do vetor entre as posições ini e fim. Use uma estratégia como a do mergesort (divisão e conquista): divida o vetor ao meio, chame a função recursivamente para cada uma das metades, depois "junte" as respostas.

2. Escreva uma função recursiva com cabeçalho

```
int maxmin (int v[], int n, int *max, int *min);
```

que devolve o maior valor do vetor em *max e o menor em *min.

3. Modifique as funções das questões anteriores para que se permita trabalhar com diversos tipos de dados utilizando templates.

4. Simule a execução do quicksort no vetor abaixo:

17 10 7 23 15 3 1 20 8 4

Indique na sua simulação as comparações que estão sendo feitas e o resultado de cada chamada do particiona. Quantas chamadas do particiona foram feitas neste caso? Quantas auto-chamadas o quicksort (recursivo) para o vetor acima geraria? Escreva, sem olhar em nenhuma anotação nem nas notas de aula, a sua versão da função particiona (ou separa). Confira seu resultado empiricamente.

5. Escreva uma função que recebe um vetor com n letras As e Bs e, por meio de trocas, move todos os As para o início do vetor. Sua função deve consumir tempo linear (proporcional ao tamanho do vetor, ou seja, a n). Demonstre empiricamente que esse é o caso.