

**Estruturas de Dados II: ed206\_ex01 quicksort**

1 – Alterar o algoritmo de *quicksort* para, ao final, mostrar o número de trocas realizadas.

2 – Dada a seqüência de números 40, 37, 95, 42, 39, 51, 60, ordene em ordem crescente utilizando o algoritmo *quicksort*, apresentando a seqüência dos números e explicando cada passo do algoritmo.

3 – Modifique o *quicksort* para fazer ordenação decrescente.

4 – Escreva e analise uma versão recursiva da função *separa*.

5 – Qual a complexidade computacional do *quicksort*? Explique.

6 - O *quicksort* produz uma ordenação estável do vetor? Explique.

7 - Submeta o vetor 77 55 33 99 indexado por 1..4 ao algoritmo *quicksort*. Teremos a seguinte seqüência de invocações da função (observe a endentação):

```
quicksort( v,1,4)
  quicksort( v,1,2)
    quicksort( v,1,0)
    quicksort( v,2,2)
  quicksort( v,4,4)
```

8 - Repita o exercício com o vetor 55 44 22 11 66 33 indexado por 1..6.

9 - Escreva uma versão não recursiva do algoritmo *quicksort*.

10 - Escreva uma versão aleatorizada (= randomized) do algoritmo *quicksort*.