

I Considere o tipo `Tree` correspondente aos nós de uma árvore binária de pesquisa de strings organizada segundo o campo `str` de cada elemento:

```
typedef struct tree{  
    char* str;  
    struct tree *left, *right;  
} Tree;
```

Implemente, na linguagem C, a função `void reverseSort(Tree *t)` que mostra no standard output os elementos da árvore binária com raiz em `t`. Os elementos devem ser mostrados de forma **inversa** à ordem alfabética do campo `str`.

Considerando uma árvore com  $N$  elementos, indique e justifique a complexidade assintótica da função implementada numa análise de pior caso.

II Considere uma árvore AVL para inteiros, inicialmente vazia, onde são inseridos sequencialmente os elementos do seguinte vector:

$\langle 5, 10, 15, 13, 14, 7, 16, 9, 17 \rangle$

Se existirem rotações durante a inserção dos elementos, ilustre as árvores antes e após cada operação de rotação, indique quais as rotações efectuadas (`rotL`, `rotR`, `rotLR` e/ou `rotRL`) e o nó sobre o qual cada operação é realizada (e.g., `rotL(44)`).

Caso não efectue qualquer rotação deverá escrever *s/ rotações*, ilustrando apenas a árvore final.

Indique a sequência de elementos visitados por uma travessia *pre-order* após a inserção de todos os elementos.