Aula prática 8

Esta aula tem como objetivo estudar a utilização de árvores AVL e Heaps. São disponibilizadas implementações de algumas operações de manipulação destas estruturas de dados e pretende-se adicionar novas funcionalidades nalguns casos de aplicação. Estes exercícios foram retirados de uma prova de avaliação de uma edição anterior da unidade curricular.

1 Tendo por base as bibliotecas de estruturas de dados apresentadas, implemente as funcionalidades pedidas nas duas alíneas seguintes no ficheiro **prob1.c**. Sempre que conveniente utilize as funções disponíveis nas estruturas árvore AVL e heap.

1.1 Implemente a função procura_inicio para uma **árvore AVL** (definida pelo nó raiz) que devolve o número de *strings* que começam com um dado carater.

```
int procura_inicio(no_avl *no, char inicio)
```

O primeiro parâmetro da função é o apontador para o nó raiz da árvore e o segundo é o carater a ser usado na pesquisa. A função deverá devolver o número de *strings* da árvore que começam por inicio.

Indique ainda num comentário no início do código da função qual a complexidade do algoritmo que implementou (não é necessário justificar).

Depois de implementada a função, o programa deverá apresentar:

```
Numero de strings comecadas por 'a': 18

Numero de strings comecadas por 'b': 19

Numero de strings comecadas por 't': 13

Numero de strings comecadas por 'z': 2
```

1.2 Implemente a função seleciona_por_ordem que retorna um apontador para o k-ésimo elemento por ordem crescente de prioridade. Depois de terminar, a função deve manter todos os elementos na heap.

```
char* seleciona_por_ordem(heap *h, int k)
```

O primeiro parâmetro é o apontador para a heap e o segundo indica qual a ordem de prioridade do valor a imprimir. Os parâmetros de entrada devem ser verificados, e a função deve retornar NULL se não for bem sucedida.

Depois de implementada a função, o programa deverá apresentar:

```
Prioridade 1: africa do sul

Prioridade 25: micronesia

Prioridade 50: moldavia
```