

U.C. 21046

**Estruturas de dados e algoritmos fundamentais**

21 de junho de 2013

### INSTRUÇÕES

Para a resolução deste **p – Fólio** aconselha-se que:

- Verifique o exemplar que lhe foi entregue e, no caso de estar incompleto ou com qualquer deficiência, dirija-se ao professor vigilante.
- O **p-fólio** é composto por 3 questões.
- O teste termina com a palavra **FIM**.
- Utilize, sempre, uma letra legível e não use uma caneta de outra cor que não seja o preto ou o azul - as respostas a lápis não serão consideradas.
- Tenha em atenção que o **p-Fólio** tem a duração de 1 hora e 30 minutos.

### Critérios de avaliação e cotação

- Deve assinalar todas as opções tomadas: no código dos seus programas, todas as constantes, variáveis, métodos ou funções devem ser devidamente explicadas através de comentário
- As respostas, que embora, sintática e semanticamente corretas, se apresentem pouco estruturadas serão severamente penalizadas, ou não consideradas.
- As respostas sem justificação serão fortemente penalizadas
- As respostas de conteúdo inadequado não serão consideradas.

### 1º Questão (4 Valores)

Considere uma árvore binária tal que a informação contida em cada nó da árvore é composta por um número inteiro e dois apontadores, respectivamente para os filhos direito e esquerdo desse nó. Assuma que cada nó da árvore é do tipo `TreeNode`, indicado à direita.

```
typedef struct _TreeNode {  
    int value;  
    struct _TreeNode *right, *left;  
}TreeNode;
```

- a) Implemente uma função que receba um apontador para a raiz de uma árvore binária com nós do tipo indicado, e que retorne a soma dos valores de todas as folhas da árvore. Utilize a seguinte assinatura para a função: `int sum_leafs(TreeNode* treeRoot);`
- b) Supondo que a árvore a que o seu algoritmo é aplicado possui  $N$  nós qual a complexidade do código desenvolvido em a) Justifique.

### 2º Questão (4 Valores)

A direção geral de viação, sendo responsável pelo bom funcionamento rodoviário, tem também como funções manter uma base de dados atualizada com informação respeitante a condutores e veículos. Cada condutor tem associada uma carta de condução, identificada por número, entidade emissora, local de emissão, data de emissão, data de renovação obrigatória e classes de veículos que o condutor está autorizado a conduzir. Cada condutor possui, ainda, uma morada e um cadastro de infrações, com data, gravidade e descrição da infração. Para os veículos, identificados univocamente pela sua matrícula, encontra-se arquivada informação acerca da marca, modelo, cor, ano de fabrico, cilindrada e situação do veículo (normal, abatido ou roubado). Pretende-se um sistema que permita efetuar as seguintes operações:

- Inserir um condutor ou um veículo;
- Remover um condutor (dado o número de carta) ou um veículo (dada a matrícula);
- Inserir uma infração no cadastro do condutor (dado o número de carta);
- Listar por ordem crescente de data, a data, gravidade e descrição das infrações de um condutor (dado o número de carta) entre duas datas;
- Listar por ordem crescente de ano de fabrico, toda a informação referente aos veículos numa determinada situação.

Assuma que o número total de infrações de um condutor é, geralmente inferior a 20, e que o número de condutores e de veículos é, à partida, ilimitado. Explícite detalhadamente as estruturas de dados mais adequadas para implementar estes sistema e descreva sumariamente como funcionariam as várias operações. Indique quais os elementos que residiriam em memória secundária e aqueles que seriam colocados em memória principal.

### 3º Questão (4 Valores)

- a) Considere o seguinte vetor de inteiros: 0 6 3 2 7 9 5 4 8 5. Represente os passos seguidos para ordenação do vetor utilizando o algoritmo Insertion Sort (ordenação por inserção).
- b) Suponha que se cria uma árvore  $B^+$  de ordem  $m$  e capacidade  $b$ , vazia, na qual se inserem os elementos 1,2,3,4,5,6,...,k, por esta ordem, com  $k$  de uma ordem de grandeza muito superior à de  $m$  e de  $b$ . Caracterize, de forma precisa e concisa, o estado de ocupação das folhas da árvore, após a inserção dos  $k$  elementos, e explique claramente a razão desse estado.

FIM