

## MC202 - 2o. sem/2008 Segunda Prova

Na solução das questões, você pode usar 'pseudo-código', como o utilizado na função de percurso em profundidade num grafo, mostrada a seguir

```
função percorre(Grafo g, Vértice v){
    marque vértice v;
    para todo vértice w adjacente a v faça
        se w não marcado então percorre(g,w);
}
```

Todas as questões têm o mesmo valor.

Questão 1: Você precisa fazer uma aplicação que trabalha com 'matrizes esparsas' : são matrizes imensas (o número de linhas e colunas é da ordem de dezenas de milhares) em que a grande maioria dos elementos é igual a zero (o número de elementos não nulos é menor que o  $\log_2$  do número de elementos da matriz). Nesse tipo de situação o que se costuma fazer é representar apenas os elementos não nulos da matriz para reduzir o uso de memória e o tempo de processamento.

1.a -Proponha uma representação para matrizes esparsas, supondo que sua aplicação vai fazer operações como soma e produto de matrizes esparsas.

1.b- Usando a representação proposta, descreva os algoritmos para soma e multiplicação de matrizes esparsas.

1.c- faça uma estimativa de tempo e memória para o seu algoritmo em função do tamanho da matriz e do número de elementos não nulos.

2-Um ciclo num grafo é um caminho da forma  $v_1-v_2-\dots-v_1$ , onde  $v_1 .. v_n$  são vértices do grafo. Escreva uma função para verificar se um grafo tem algum ciclo. Faça uma estimativa do uso de tempo e memória do seu algoritmo.

3-Escreva uma função para verificar se existe num certo grafo o caminho  $v_1-v_2-\dots-v_k$ , onde  $v_1 .. v_k$  são vértices desse grafo (passados como parâmetro da sua função). Faça uma estimativa do uso de tempo e memória necessários à execução da mesma.

Para as questões 4 e 5, considere uma árvore binária formada por nós do seguinte tipo:

```
typedef struct no *apno;
typedef struct no { int info; apno esq, dir, pai; } no;
```

4- Alguns tipos de árvores balanceadas de busca exigem que todas as folhas tenham a mesma altura. Escreva uma função que determine se uma árvore binária atende a essa exigência. Faça uma estimativa de tempo e memória necessários à execução da mesma.

5 – Escreva uma função para percorrer uma árvore binária em pré-ordem, não recursiva e sem usar uma pilha.