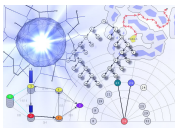


Frequência 2

Atenção: Para todos os exercícios do teste podem ser implementadas funções auxiliares, desde que transcritas para a folha de prova.

1. Considere a seguinte estrutura de dados binária:

```
typedef struct NODO{
        int id;                    %critério de ordenação na árvore
        char nome[80];
        struct NODO *fe;
        struct NODO *fd;
}Nodo;
```

   a. (4 valores) Implemente uma função que verifique se dois nós (especificados através de apontadores) pertencem ao mesmo ramo da árvore binária (1=sim, 0=não).

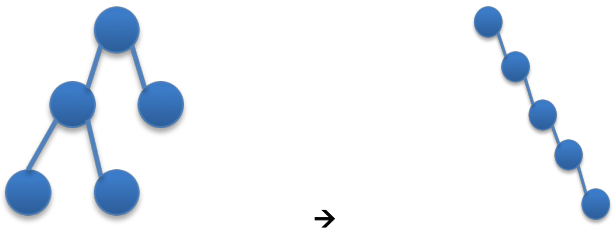
   Protótipo: int igualRamo(Nodo* A, Nodo *n1, Nodo *n2);
   //A=raiz, n1,n2=nós a pesquisar

   b. (3 valores) Implemente uma função que devolva o total de nós descendentes de determinado nó da árvore, especificado através do nome. Considere que tem à sua disposição a função "int contaNós(Nodo *A)", que retorna o total de nós de uma árvore binária.

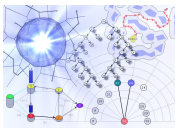
   Protótipo: int totalDescendentes(Nodo* A, char *nome);
   //A=raiz.

   c. (4 valores) Codifique a função "**inútil**". Esta função recebe uma árvore binária e transforma-a numa estrutura de acesso sequêncial, em que todos os nós apenas têm apenas ligação ao filho direito.

   Exemplo:



   Protótipo: Nodo* inutil(Nodo* A);
   //A=raiz



2. (3 valores) Apresente o esquema de uma árvore binária não-balanceada, em que duas operações de rotação à esquerda a tornem balanceada.

3. (3 valores) Implemente uma função que devolva a profundidade de um determinado nó na árvore (especificado através do ID). A função deverá devolver -1 caso o nó não seja encontrado.

   Protótipo: int profundidadeNo(Nodo* A, int id);
   //A=raiz, id=nó a procurar

4. (3 valores) Validade de caminhos. Considere que um "caminho" na árvore (formado pelos nós {a,b,c}) existe quando for possível aceder directamente ao nó "b" a partir de "a" e ao nó "c" a partir de "b". Implemente uma função que determine se um caminho (especificado através de uma sequência de IDs) é válido. A função deverá devolver 1 em caso afirmativo ou 0 caso contrário.

   Protótipo: int caminhoVálido(Nodo* A, int *ids, int tot);
   //A=raiz, ids=vector com os ids a procurar, tot=total de elementos no vector de ids.