

Utilizando as funções do TAD Arbin, Lista, Pilha e Fila resolva as questões abaixo:

-nomenclatura: Arbin -> Árvore Binária

1) int pesoArbin( Arbin a){...}

Calcular e retornar o peso de uma árvore binária ( número de elementos da árvore).

**Obs: a complexidade desta função é  $O(N)$**

2) int estaArbin( Arbin a, TipoA elem){...}

Verificar se um elemento está presente em uma árvore binária.

**Obs: a complexidade desta função é  $O(N)$  se a árvore estiver degenerada e  $O(\log N)$  se a árvore estiver balanceada(cheia).**

3) int numFolhas( Arbin a){...}

Calcular o número de folhas de uma Arbin.

**Obs: a complexidade desta função é  $O(N)$**

4) int numOcorrencias( Arbin a){...}

Calcular o número de vezes que um elemento aparece na Arbin.

**Obs: a complexidade desta função é  $O(N)$**

5) int existeCaminho( Arbin a, TipoA e1, TipoA e2){...}

Verificar se existe um caminho entre dois elementos e1 e e2 de uma Arbin.

**Obs: a complexidade desta função é  $O(N)$**

6) int contNivel( Arbin a, int nivel){...}

Calcular o número de elementos de uma Arbin em um dado nível.

**Obs: a complexidade desta função é  $O(N)$**

7) int semelhantesArbin( Arbin a1, Arbin a2){...}

Verificar se duas Arbin a1 e a2, sem elementos repetidos, são semelhantes.

Utilizar duas funções:

-A) a primeira função(semelhantesArbin) verifica se as duas Arbin têm o mesmo peso, e que todos os elementos da primeira árvore estejam na segunda árvore(é a segunda função descrita em B).

-B) a segunda função verifica, para cada elemento de a1, se este está presente em a2.

**Obs: a complexidade desta função é  $O(N*M)$ , onde N é o peso de a1 e M é o peso de a2.**

8) Desenhe duas árvores isomorfas e semelhantes, que não sejam iguais.

9) Determine os seguintes valores para uma árvore binária:

- a) Número mínimo e máximo de elementos de uma árvore completa de N níveis
- b) Número mínimo de níveis de uma árvore binária com peso P.
- c) Número máximo de folhas de uma árvore binária com N níveis.
- d) Número mínimo e máximo de elementos presentes no nível N de uma Arbin completa de altura H.
- e) Número de elementos de uma Arbin cheia de N níveis

10) int alturaArbin( Arbin a){...}

Calcular a altura de uma Arbin.

11) int iguais(Arbin a1, Arbin a2){...}

Verificar se duas Arbin a1 e a2 são iguais.

12) int isomorfos(Arbin a1, Arbin a2){...}

Verificar se duas Arbin a1 e a2 são isomorfas.

13) int completaArbin(Arbin a){...}

Verificar se Arbin a é completa.

14) int cheiaArbin(Arbin a){...}

Verificar se Arbin a é cheia.

15) Lista buscaCaminhoArbin(Arbin a, TipoA elem){...}

Retorna a lista de elementos correspondente ao caminho que vai da raiz da Arbin a, até o elemento elem. Se não existe o caminho, retorna uma lista vazia.

16) int ocorreArbin(Arbin a1, Arbin a2){...}

Verifica se a Arbin a2 ocorre na Arbin a1.

17) Reconstruir a Arbin com os seguintes caminhamentos:

a)

pre-ordem: 10-20-30-50-60-40-70-80-90

in-ordem: 50-30-60-20-80-70-90-40-10

b)

pre-ordem: 60-30-80-70-40-20-50-90-10

in-ordem: 30-60-20-80-70-40-10-90-50

18) int nivelArbin(Arbin a1, TipoA elem){...}

Calcular o nível em que aparece o elemento elem, em uma Arbin sem elementos repetidos. Se o elemento não está presente, a função retorna -1.

19) int maiorElementoArbin(Arbin a){...}

Utilizando um caminharmento recursivo In-ordem, esta função retorna o maior elemento da Arbin a.