

Tipos Abstractos de Dados Árvore Binária

41 Implementação do TAD Árvore Binária.

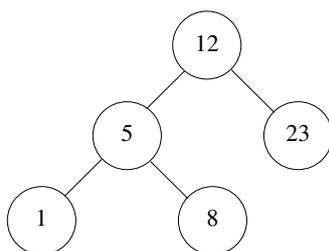
$$AB = (\{Abvazia, ABesq:elemento:ABdir\}, \{criar,destruir,insserir, retirar,procuraElemento,vazia?\})$$

considerando que a inserção é feita de forma a criar uma árvore binária de pesquisa, isto é, à esquerda do elemento na raiz, todos os elementos são menores do que a raiz e à direita, todos os elementos são maiores ou iguais à raiz. Implemente o TAD Árvore Binária (de pesquisa).

42 Travessias de árvores

Ao contrário das estruturas de dados lineares (vectores, listas, filas, pilhas, etc.), que possuem apenas uma maneira lógica de serem percorridas, as árvores podem ser percorridas de várias formas diferentes.

As formas mais usuais para travessias de árvores são: em-ordem, pré-ordem, pós-ordem, travessia em largura.



- Travessias em profundidade:
 - Em-ordem (Esquerda, Raiz, Direita): 1 5 8 12 23;
 - Pré-ordem (Raiz, Esquerda, Direita): 12 5 1 8 23;
 - Pós-ordem (esquerda, direita, raiz): 1 8 5 23 12.
- Travessia na largura (por níveis): 12 5 23 1 8.

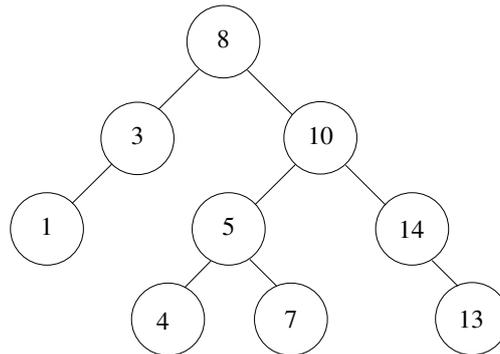
Implemente as diferentes travessias em profundidade.

43 Seja ABI o tipo abstracto de dados *Árvore Binária de Inteiros*, em cujos nós se armazena um número inteiro e assuma que os números inteiros não se encontram registados em nenhuma ordem particular.

Elabore sub-programas para:

1. Escrever os elementos registados numa árvore binária *em-ordem* (*pós-ordem* / *pré-ordem*).
2. Contar o número de nós de uma árvore binária.
3. Listar os elementos registados nas folhas de uma árvore.
4. Contar o número de folhas de uma árvore binária.

5. Contar os nós internos de uma árvore binária.
 6. Obter um ponteiro para o nó que contém o maior elemento registado numa árvore.
 7. Calcular a altura de uma árvore.
 8. Construir uma árvore binária completa com profundidade n , sendo os nós gerados recursivamente utilizando pré-ordem.
- 44 Árvores de pesquisa é uma árvore binária em que, para cada nó, temos as seguintes propriedades:



- A sub-árvore esquerda de um nó contém apenas nós com valores menores que o valor do nó.
 - A sub-árvore direita de um nó contém apenas nós com valores maiores ou iguais que o valor do nó.
 - As sub-árvores esquerda e direita também devem ser uma árvore binária de pesquisa.
1. Implemente uma função de construção de uma árvore binária de pesquisa a partir de uma lista de elementos.
 2. Implemente uma função de pesquisa binária utilizando a estrutura de árvore binária de pesquisa.

45 Seja *ABI* o tipo abstracto de dados *Árvore Binária de Inteiros*, em cujos nós se armazena um número inteiro e assuma que os números inteiros não se encontram registados em nenhuma ordem particular.

Elabore sub-programas para:

1. Dados uma árvore e dois inteiros, a, b , verificar se b é **descendente** de a , isto é, se b pertence a uma das sub-árvores do nó que contém a . NOTA: Deve ser contemplada a possibilidade de a e/ou b não pertencerem à árvore dada.
2. Dado uma árvore binária e um inteiro, escreva todos os nós descendentes (ascendentes) do nó correspondente ao inteiro dado.

46 O *percurso em largura* de uma árvore binária consiste em percorrer os nós de uma árvore por níveis, desde o mais alto (raiz) até à base e da esquerda para a direita.

Por exemplo, para a árvore da figura 2 ter-se-ia **ABODFGEMC**.

Elabore um sub-programa para fazer uma travessia em largura.

Sugestão: utilize uma lista linear para indicar as sub-árvores ainda a serem percorridas.

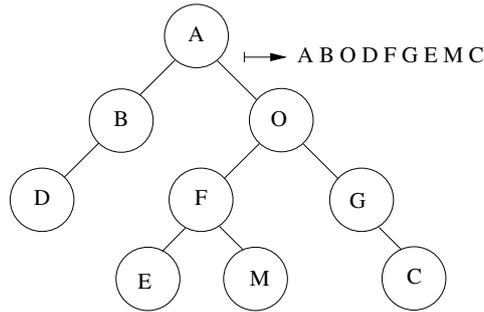


Figura 2: Árvore Binária — Percurso em Largura

47 A igualdade entre árvores binárias pode ser entendida como uma igualdade entre listas de elementos após a travessia das duas árvores binárias (por exemplo, em ordem). Pode também ser entendidas como uma igualdade estrutural, caso em que se pretende que além de elementos iguais as árvores tenham uma estrutura igual.

Temos então as seguintes definições:

Igualdade nos elementos – dados duas árvores binárias, ab_1 e ab_2 elas dizem-se iguais (nos elementos) se $travessia(ab_1) = travessia(ab_2)$.

Equivalência – dados duas árvores binárias, ab_1 e ab_2 elas dizem-se equivalentes, $eq(ab_1, ab_2)$, se: ambas as raízes forem vazias ou $obtemRaiz(ab_1) = obtemRaiz(ab_2)$ e $eq(obtemEsq(ab_1), obtemEsq(ab_2))$ e $eq(obtemDir(ab_1), obtemDir(ab_2))$.

Elabore sub-programas para:

1. Determinar se duas árvores binárias são iguais (nos elementos).
2. Determinar se duas árvores binárias são equivalentes.

48 Dadas duas árvores binárias, ab_1 e ab_2 diz-se que ab_1 está contida em ab_2 se ab_1 é equivalente a ab_2 ou então existe pelo menos uma sub-árvores de ab_2 que seja equivalente a ab_1 .

Elabore um sub-programa para:

1. Determinar se uma árvores binárias ab_1 está contida numa outra árvore binária ab_2 .

49 Pretende-se guardar a informação sobre as diferentes unidades de ensino de uma dada universidade, preservando a estrutura hierárquica da mesma. Por exemplo, na figura 3, tem-se uma representação parcial da estrutura da Universidade de Coimbra.

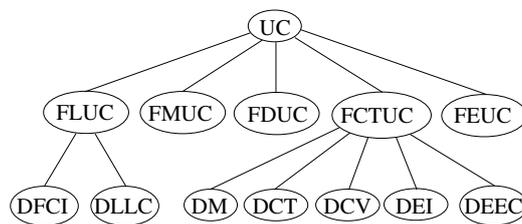


Figura 3: Estrutura Hierárquica — Árvore Genérica

Elabore sub-programas para:

1. Inserir a informação, na forma de uma árvore binária (sem que haja perda de informação quanto à estrutura hierárquica).
2. Que, para uma dada faculdade, liste todos os seus departamentos.
3. Que liste todas as faculdade, conjuntamente com os respectivos departamentos.