

Avaliação árvores binárias I

Estrutura de Dados

Abril de 2022

Orientações:

- (a) Essa é uma avaliação sobre árvores binárias, referente às aulas 6, 7, 8 e 9 do AVA.
- (b) A data de entrega poderá ser feita até o dia 01.05.
- (c) O fórum "Avaliação árvores binárias" correspondente a esta atividade será disponibilizado para esclarecimento de dúvidas.
- (d) Não deixe para a última hora.
- (e) as funções solicitadas podem ser escritas em pseudo linguagem conforme feito em aula ou usando a linguagem Python, C++, C ou Java. Entretanto, a estrutura deve ser a mesma daquela definida em sala de aula. Se você construir métodos ou funções auxiliares, você deve apresentá-los explicitamente.
- (f) A avaliação é individual.

-
1. Desenhe a árvore binária com a seguinte representação em parênteses.

$$\left(\left((C)I(H) \right) F \left(((G)A(D(B)))E \right) \right)$$

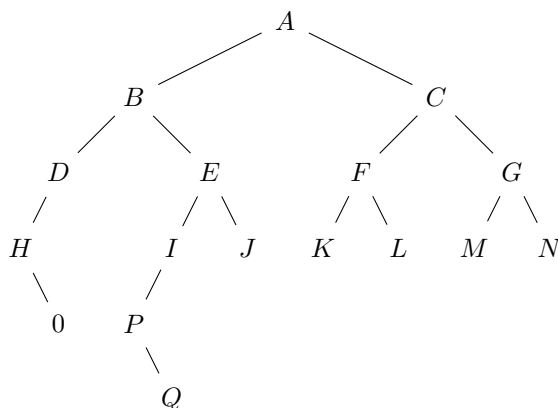
2. Escreva uma função que recebe os nós u e v e devolve `true` se u é descendente de v .
3. Escreva uma função que devolve a altura $h(u)$.
4. Desenhe uma árvore binária que possui a seguinte ERD e a seguinte RED.
ERD: MHBGDCIELAJKF
RED: EHMCGBDIKJLAF

5. Desenhe uma árvore binária que possui a seguinte ERD e a seguinte RED.

ERD: CGKBJFIAEDH

RED: HGCKIJBFEAD

6. Desenhe uma árvore binária de busca inicialmente vazia, após a inserção da seguinte sequência de chaves (sem rotação) usando (obrigatoriamente) o algoritmo visto em aula: 7, 10, 15, 3, 1, 2, 6, 9, 0.
7. Desenhe a árvore binária de busca a seguir (abaixo) após a remoção da seguinte sequência de chaves (sem rotação) usando obrigatoriamente o algoritmo visto em aula: B, D, C



8. Desenhe uma árvore binária de busca inicialmente vazia, após a inserção da seguinte sequência de chaves (sem rotação) usando obrigatoriamente o algoritmo visto em aula. Além disso, após as inserções: 10, 3, 7, 2, 11, 0, 1, 6, 8. Após todas as inserções, diga qual é o fator de balanceamento $b(u)$ de cada nó u .
9. Escreva uma função f que recebe a raiz r de uma árvore binária T e imprime o campo **info** e a altura de cada nó de T . Seu algoritmo deve gastar necessariamente tempo $O(n)$.
10. Escreva uma função **ehDeBusca** que recebe a raiz r de uma árvore binária T e devolve **true** se T é árvore de busca e devolve **false** caso contrário.

Obs: Lembre-se que os campos de um nó são: **esq**, **dir**, **pai** e **info**, onde, nesse último está armazenada a chave usada para a busca. Se isso ajudar, você pode assumir que estão implementadas as funções **primeiro(u)** e **ultimo(u)** que recebem um nó u e devolvem o primeiro e o último nós respectivamente em uma busca ERD.