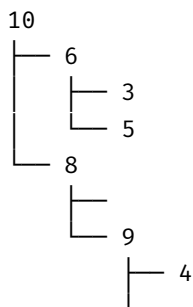


Busca e árvores - Prática

Marco A L Barbosa

malbarbo.pro.br

- 1) Considere um arranjo com os valores 3, 6, 8, 9, 20, 21, 22, 23, 30, 40, 45, 46, 50, 60 e diga quais são os índices dos valores que são comparados com a chave em uma busca binária para cada uma das chaves: 6, 30, 41, 50, 70.
- 2) Implemente a busca binária de forma que a chave seja comparada por igualdade com o elemento “do meio” apenas uma vez. Dica: Faça a repetição até que `inicio == fim` e faça a comparação após a repetição.
- 3) Crie um experimento para verificar qual das duas implementação de busca binária é mais eficiente: a vista em sala ou a do exercício anterior.
- 4) Implemente o TAD Dicionario usando arranjo ordenado e busca binária.
- 5) Desenhe uma árvore binária com os elementos 3, 6, 8, 20, 21, 22, 23, 30, 40, 46, 50, 60 de maneira que qualquer busca binária em um arranjo com esses elementos ou na árvore se comporte da mesma forma (como o exemplo do material).
- 6) Projete uma função que determine a quantidade de elementos de uma árvore binária.
- 7) Projete uma função que determine quantos nós em uma árvore binária tem grau 2.
- 8) Uma árvore binária cheia é aquela em que todos os seus nós tem grau 0 ou 2. Projete uma função que determine se uma árvore binária é cheia.
- 9) Projete uma função que determine a altura de uma árvore binária.
- 10) Projete uma função que altere os elementos negativos de uma árvore binária para seus valores absolutos.
- 11) Projete uma função que devolva o valor máximo em uma árvore binária ou `None` se a árvore estiver vazia.
- 12) Uma árvore binária balanceada é aquela em que a altura das subárvores a direita e a esquerda diferem em no máximo 1 e as duas subárvores também são balanceadas. Projete uma função que verifique se uma árvore binária é balanceada.
- 13) (Desafio) Projete uma função que exiba uma árvore na horizontal, como no exemplo a seguir. Note que o nó 8 não filho a esquerda e que o nó 9 não tem filho a direita. Dica: Faça uma função recursiva auxiliar (`linhas`) que devolva uma lista de strings, onde cada string representa uma linha a ser exibida, para exibir a linhas faça `print('\n'.join(linhas(t)))`.



- 14) Projete uma função que determine se uma árvore binária é uma árvore binária de busca (ABB).
- 15) Projete uma função que encontre a amplitude dos valores de uma ABB não vazia. A amplitude de valores é a diferença entre o valor máximo e mínimo. A sua função deve ter tempo de execução $O(h)$,

onde h é a altura da árvore. Dica: faça duas funções auxiliares, uma para encontrar o mínimo e outra para encontrar o máximo.

- 16) (Desafio) Projete uma função que receba como parâmetro uma ABB e um valor n , contido na ABB, e determine o sucessor de n na ABB, isto é, o menor valor maior que n . A função deve devolver `None` se não existe sucessor de n .
- 17) Projete uma função que crie um arranjo ordenado em ordem crescente a partir dos valores de uma ABB.
- 18) Projete uma função que crie um arranjo ordenado em ordem decrescente a partir dos valores de uma ABB.
- 19) (Desafio) Projete uma função que crie um arranjo com os valores de uma ABB em ordem de nível da direita para esquerda.
- 20) Projete uma versão iterativa da operação de inserção em ABB.
- 21) Projete uma versão iterativa da operação de remoção em ABB.
- 22) Desenhe a sequência de árvores AVL geradas pela inserção dos elementos 6, 1, 3, 7, 8, 12, 9, 5, 4, 2, 16.
- 23) Partindo do resultado do exercício anterior, desenhe a sequência de árvores AVL geradas pela remoção dos elementos 12, 1, 3, 6, 7, 8, 5, 9, 4, 2.
- 24) Baixe o arquivo `avl.py` e faça a implementação das funções `rotaciona_dir` e `balanceia_dir`. Depois modifique a função `remove` adicionando as chamadas para fazer o rebalanceamento.
- 25) Implemente o TAD `Dicionario` usando árvores AVL. Note que você terá que alterar o arquivo `avl.py` para armazenar o valor associado com a chave no nó.
- 26) O Alberto é um estudante de Computação e quando estava estudando árvore AVL teve a ideia de criar um novo tipo de árvore balanceada: limitar a diferença entre a quantidade de nós a esquerda e a direita da raiz em 1 nó. Discuta os méritos da ideia do Alberto e as diferenças na implementação em relação a árvore AVL.