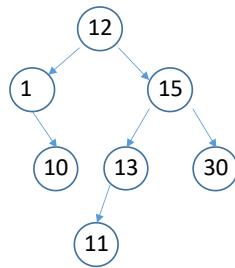
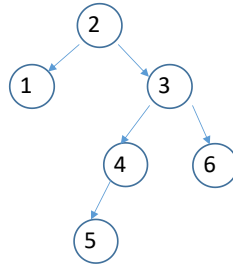


Lista de Exercícios Árvore Binária de Busca (ABB)

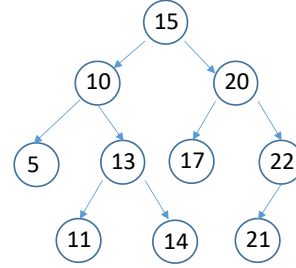
1) Verificar se as árvores indicadas a seguir são árvores binárias de busca:



(a)



(b)



(c)

2) Desenhar uma ABB construída pelo algoritmo de inserção, supondo a seguinte ordem das chaves: S3, S7, S1, S2, S6, S5, S4, onde $S = \{S1, S2, S3, S4, S5, S6, S7\}$ e $S_i < S_{i+1}$, $i < 7$.

3) Provar ou dar contra-exemplo:

Sejam P_1 e P_2 duas permutações de um conjunto de chaves S e T_1 e T_2 as ABB's correspondentes a P_1 e P_2 , respectivamente. Então $P_1 \neq P_2$ se e somente se $T_1 \neq T_2$.

4) Seja $S = \{S1, \dots, S7\}$ um conjunto de chaves $S_i < S_{i+1}$, $i < 7$. Desenhar ABB com as seguintes propriedades:

- T possui altura máxima.
- T possui altura mínima.

5) Escreva em C uma função que recebe como parâmetro o endereço do nó raiz de uma ABB e um número inteiro. A função deve retornar o endereço do nó da chave imediatamente maior do que a chave dada. Se a chave não pertencer à árvore a função deve retornar NULL.

6) Dada uma ABB,

- Inserir, para cada nó, a sua posição relativa na sequência ordenada de forma crescente.
- Construir algoritmo de busca pela posição relativa.

Observação: Se as chaves em ordem crescente são: 10, 30, 50, 60, 80, 100 a posição relativa da chave 10 é 1ª, da 30 é a 2ª, da 50 é a 3ª e assim sucessivamente.

7) Indicar Falso ou verdadeiro, justificando.

- A complexidade da exclusão em ABB é $O(\log N)$.
- A complexidade da busca de uma dada chave em ABB é $O(\log N)$.
- A complexidade da busca de uma dada chave em uma árvore binária é $O(N)$.

8) Escrever os algoritmos de busca e inserção em ABB.

9) Escrever o algoritmo de remoção em ABB.