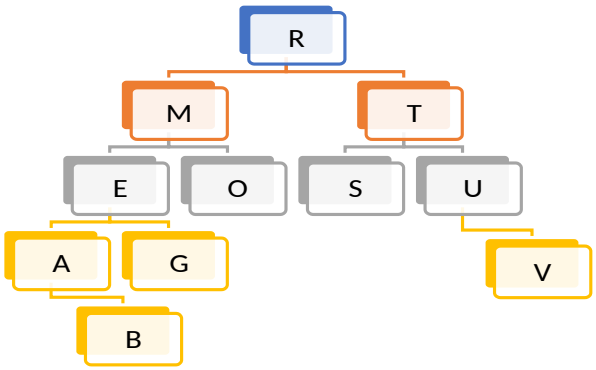
1.
   a. Draw the binary search tree that is created if the following numbers are inserted in the tree in the given order.

      12, 15, 3, 35, 21, 42, 14
   b. Circle all the leaves. Put a square box around the root. Draw a star around each ancestor of the node that contains 35. Put a big X through every descendant of the node the contains 35.
   c. Write the node labels in the order they would be printed in an in-order traversal of the tree.

2. List all of the nodes of height 2 in the below tree. Is the below tree a binary search tree? Why or why not?



3. Consider the binary search tree in Question 2. Perform the following operations. For each operation, start with the original tree.
    a. Insert H
    b. Insert W
    c. Delete U
    d. Delete R

4. Implement a) the **minimum** method in the following BST class, which finds the smallest value in a BST.

```
public class BST{
     int value;
     BST left;
     BST right;

     /** Returns the minimum value in BST n.
       * pre: n is not null */
     public static int minimum(Node n) {
          //TODO









   }

}
```

5. The **n** nodes in a binary tree can be visited in:
    a. O(1) time
    b. O(logn) time
    c. O(n) time
    d. O(n logn) time
    e. $O(n^2)$ time

6. What is the maximum height of a binary tree containing n nodes?