**Practice Problems on TREES, AVL and HEAPS**

**Q1**: What is the maximum number of nodes in a binary tree of height *k*. Explain your answer.

Q2: Write Recursive as well as Iterative functions for the following:

* + Function to count the number of non-leaf nodes
  + Function to count the number of leaves
  + Function to count the number of right children

Q3:Write a function that checks whether a binary tree is perfectly balanced

Q4: Write a function bool IsComplete(). The function should check whether the tree is complete or not.

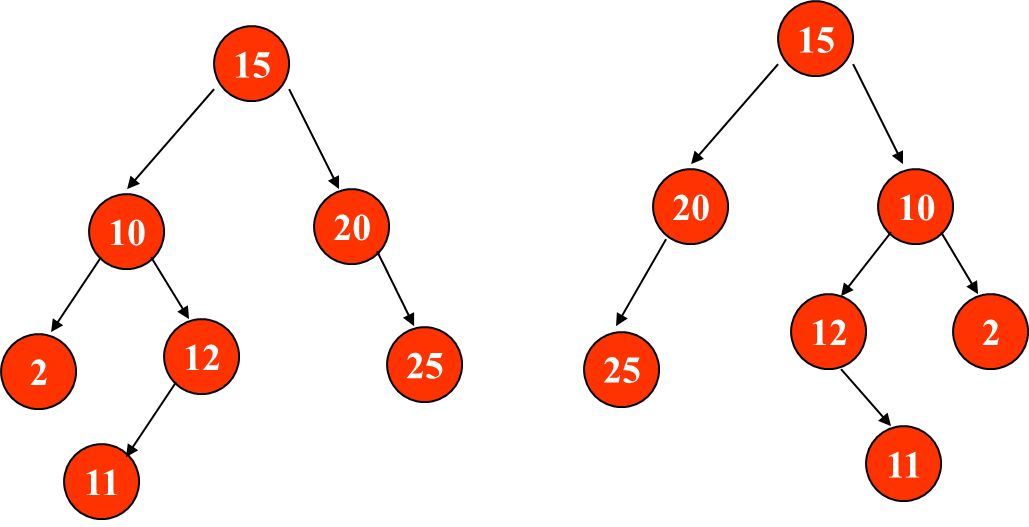
Q5: Divide a binary search tree into two trees, one tree with key < *K* and the other tree with keys ≥ *K*, where *K* is any key in the tree.

Q6: Design an algorithm to test whether a binary tree is a binary search tree

* + HINT: Test condition at each node and also keep track of upper and lower limit for the subtree

Q7: A *full node* is a node with two children. Prove that the number of full nodes plus one is equal to the number of leaves in a nonempty binary tree

Q8: Write a function to flip the BST



Q9: Write a function to print the BST in descending order

Q10: Find total number of nodes in a BST that are perfectly balanced.

A node is said to be perfectly balanced if it has the same number of left and right children

HEAPs

1. Show the result of inserting 10, 12, 1, 14, 6, 5, 8, 15, 3, 9, 7, 4, 11, 13, and 2,  
   one at a time, into an initially empty binary heap.
   1. b. Show the result of using the linear-time algorithm to build a binary heap using the same input.
   2. Show the result of performing three deleteMin operations in the heap
2. Find second min in min heap
3. Find third min in a min heap
4. Convert min heap to max heap
5. How to determine if the given array is a binary heap
6. Check if the binary tree is a binary heap
7. Show the following regarding the maximum item in the Minheap:  
   a. It must be at one of the leaves.  
   b. There are exactly *N/*2 leaves.  
   c. Every leaf must be examined to find it
   1. How much time do you need to visit all leaf nodes to find maximum?
8. Assume both heaps are perfect binary  
   trees, containing 2l - 1 and 2r - 1 nodes, respectively.  
   a. Give an O(log N) algorithm to merge the two heaps if l = r.  
   b. Give an O(log N) algorithm to merge the two heaps if |l - r| = 1.  
   \* c. Give an O(log2 N) algorithm to merge the two heaps regardless of l and r