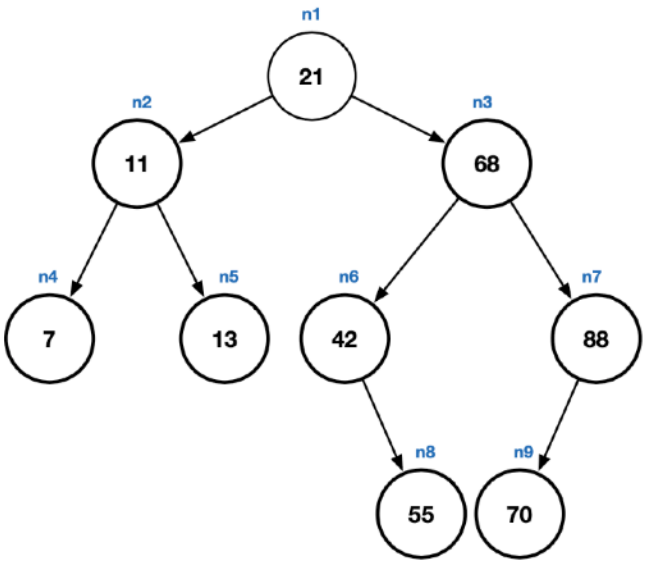
Binary Trees Lab : Binary Nodes Manipulation – Binary Search

Consider the following Binary Search Tree (BST) composed of 9 BNodes (i.e., n1 – n9)



1. (10 pts) Without code, establish the connectivity between all the nodes

2. (30 pts) **Without code**, provide:

   - Pre-Order :
   - In-Order :
   - Post-Order :

3. Now it is time to code. Using Dr.Java, construct a tree from previous questions or any other tree, and implement the following methods for BST:

- (20 pts) Write a method int countParentsWithOneChild(BNode r) that returns the number of BNodes that have only 1 child as dependency (can be left or right) in a binary search tree, where the parameter r represents the current root for the current tree.
- (20 pts) Write a method called int findTheLargest(BNode r) that returns the largest element in a binary search tree, where the parameter r represents the current root for the current tree.

4. (20 pts) Show the result of inserting following values into initially empty BST (one by one). Do not forget to consider properties of BST:

7, 8, 1, 5, 29, 17, 6, 2

What to submit:

- MyTree.java that contains at least constructor, countParentsWithOneChild(), and findTheLargest()
- MyTreeTester.java - Tester class with main to test your program.
- BNode. Java - Node class with constructors
- Word or pdf file with answers for questions 1, 2, 4