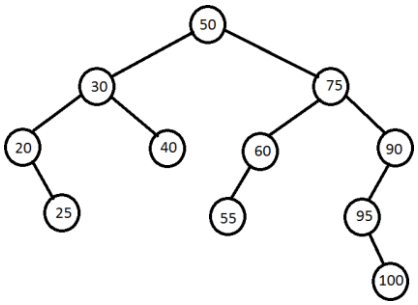
1. Tree Traversal: (40 points:)



(a) (10 points:) Given the image above, what is the level-order traversal of the tree?

50 => 30 => 75 => 20 => 40 => 60 => 90 => 25 => 55 => 95 => 100

(b) (10 points:) Given the image above, what is the pre-order traversal of the tree?

50 => 30 => 20 => 25 => 40 => 75 => 60 => 55 => 90 => 95 => 100

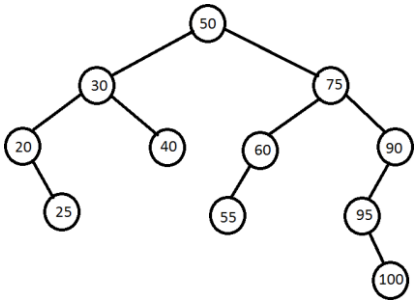
(c) (10 points:) Given the image above, what is the in-order traversal of the tree?

20 => 25 => 30 => 40 => 50 => 55 => 60 => 75 => 95 => 100 => 90

(d) (10 points:) Given the image above, what is the post-order traversal of the tree?

25 => 20 => 40 => 30 => 55 => 60 => 100 => 95 => 90 => 75 => 50

2. Binary Search Tree: (40 points:)

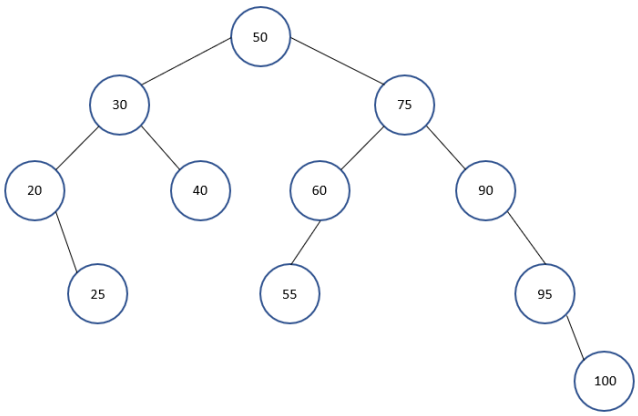


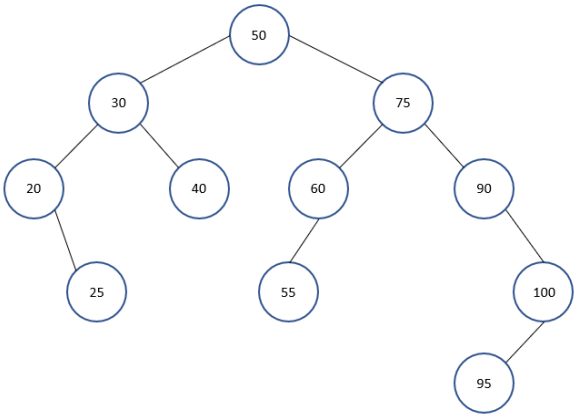
In all the questions below, nodes will be referred to by its key field. So, the root is referred to as node 50.

(a) (10 points:)
Yes/No: Does the tree above have the properties of a binary search tree?

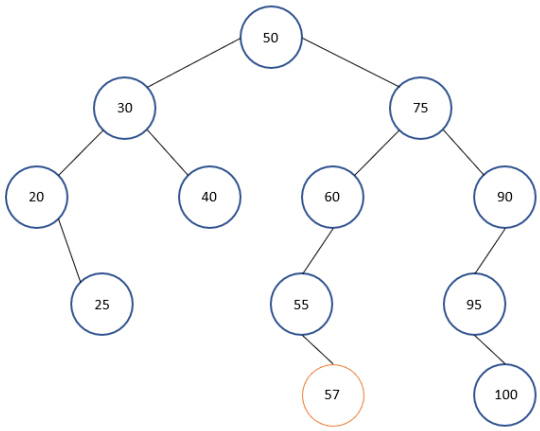
No

(b) (10 points:)
If not, redraw the tree so that it is a proper binary search tree. Make the minimal amount of change necessary.





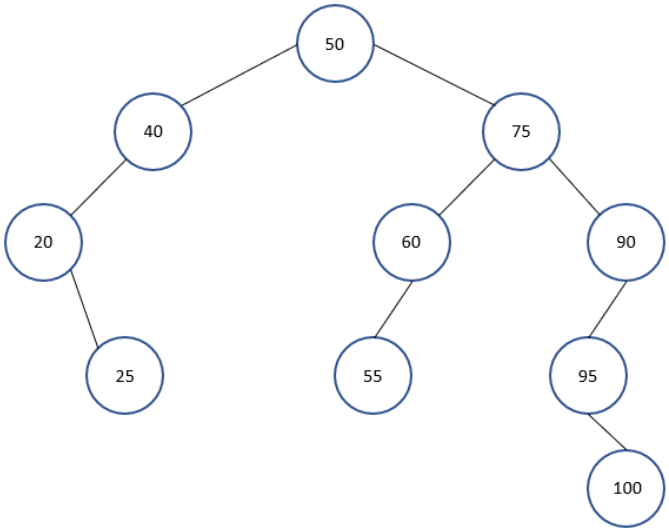
(c) (10 points:)
Draw where node 57 will be inserted on the original figure above.



(d) (10 points:)
Which node is the successor of node 50?

55

(e) (Bonus: 5 points:)
Using the space below, draw the tree after node 30 is deleted.



3. QuadTree: (20 points:)

The image above is 400x400. The different shapes vary in size from 100x100, 100x200, 300x300, 200x100, and 100x50 (width x height). Using the convention in your program for ordering quadtree children (NW, NE, SW, SE), draw the quadree representation of the image. Label the leaf nodes as R (red), G (green), or B (blue). Note: keep subdividing the tree if a quadrant is not a uniform color. You don't need to specify top-left and bottom-right coordinates.