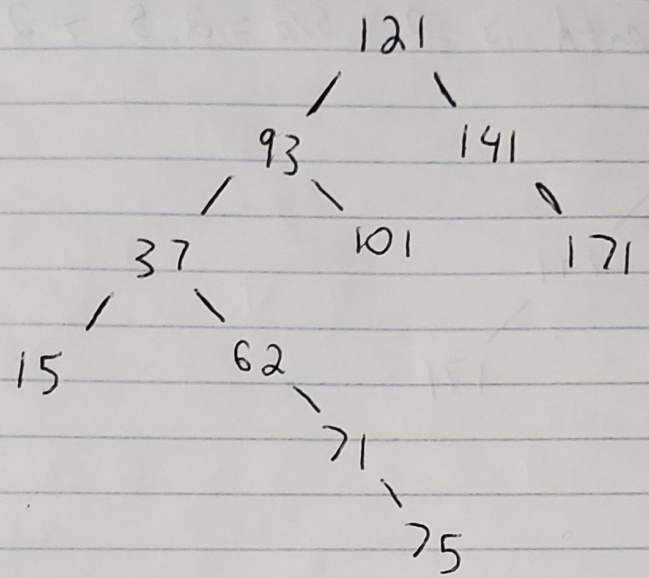
Question 3

a) 15, 37, 62, 71, 75, 93, 101, 121, 141, 171



b) One solution would be in order traversal. Since the in-order traversal returns the elements sorted in ascending order. We are essentially reversing the traversal, as we start with the list & end with the binary tree.

We start with a sorted list and a binary tree with its shape predetermined. ~~Navigate~~ Navigate the binary tree using in order traversal and slot in the elements as we visit each node.

c)

i) 121, 93, 37, 15, 62, 71, 75, 101, 141, 171

ii) First sort the list in ascending order, the in-order traverse the tree, inserting ~~one~~ an element for each node visited. This is the exact algorithm used in part b). ~~After~~ After we use pre-order traversal on the tree, which gives us the final list.

d) This cannot be a RB tree because for any given node, the ratio between the longest & shortest path must be ≤ 2. Take the root node, the longest path is 5 while the shortest path is 2 $\quad 5/2 = 2.5 > 2$.

e) Original tree