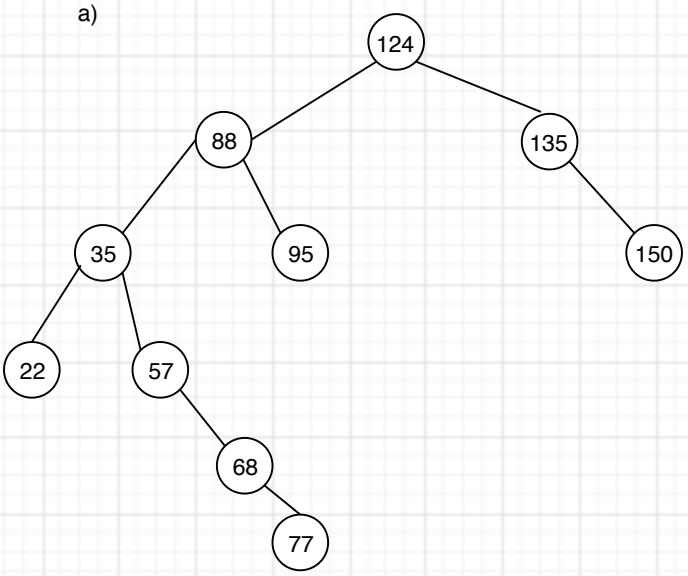
Minh Durbin
Q3

a)



b) In terms of the specific BST provided, the right subtree of the root has only 2 nodes. However the left subtree has 7 nodes. So we need to find the key in which there are only 2 numbers larger than it and 7 numbers that are smaller than it. This key is 124 and it will be the root. To fill the nodes in the most efficient way, we want to fill all nodes in a path.So starting from the root find its successor, which is 135 and put it as 124's right child. Next find 135's successor, which is 150, and put it as 135's right child. To find 124's left child, we must find 124's predecessor, which is 95. Then we find 95's predecessor, which is 88, and put 88 as 124's left child. Since we already know 88's successor, which is 95, we put 95 as 88's right child. So far 2 out of 4 paths are filled. Next we find 88's predecessor, which is 77, and put it as the leaf node in the longest path. Then find 77's predecessor, which is 68, and put it as 77's parent. Then find 68's predecessor, which is 57, and put it as 68's parent. Then find 57's predecessor, which is 35, and put it as 57's parent. The longest path has been filled. Finally we find 35's predecessor, which is 22, and put it as 35's left child.

c) The ordered list in which the keys are inserted starting at root is 124, 135, 150, 88, 95, 35, 57, 68, 77, 22. Since there are 7 nodes in the left subtree and 2 nodes in the right subtree of the root, 124 will be the root and is the first number to be inserted. Then for each path, we find the current node's predecessor and insert it starting at the leaf node of each path.

d) This tree can't be colored to form a valid RB Tree because this tree is not balanced and the longest path has twice the number of nodes as the other paths. For the paths with 3 nodes, the minimum black height is 2. For the path with 4 nodes, the minimum black height is 2. But for the longest path which has 6 nodes, the minimum black height is 3. So for the paths with 3 or 4 nodes, they satisfy the RB Tree black height property. But for the longest path, the minimum black height is 3 which does not match the other paths's minimum black height of 2. For this to be a valid RB Tree, the tree must be balanced based on the AVL tree property where the difference between heights of left and right subtrees cannot be more than one for all nodes.

e) To turn this into a valid RB Tree, the rotations must produce a tree where the difference between heights of left and right subtrees cannot be more than one for all nodes. For the first rotation, we rotate left starting at node 35 with respect to node 57. So node 35 and its children will be the left subtree of node 57. Node 57 and its subtrees will be the left subtree of node 88. Then we rotate right starting at node 124 with respect to node 88. Node 124 and its children will be node 88's right subtree and 88's right child will become node 124's left child.