2. (25 Points) Consider the following binary tree:



(a) Traverse the tree using preorder, inorder, and postorder traversal and mention the output

sequence in each of the cases.

Preorder Traversal: node,left,right

10,8,5,3,5,2,1,7,1,6

Inorder Traversal: left,node,right

3,5,5,8,1,2,10,1,7,6

Postorder Traversal: left,right,node

3,5,5,1,2,8,1,6,7,10

(b) How many internal nodes are there? What is the size and height of the tree?

Internal Nodes: nodes which are not leaves

The number of internal nodes = 7

Size of the tree: the number of node in the tree

Size = 10

Height of the tree: the depth of the deepest leef

Height = 3

(c) What is the maximum width and diameter of the tree?

Maximum width of the tree: the maximum number of the nodes at the same depth

Maximum width = 4

Diameter of the tree: the longest path between any two nodes

Diameter = 5

(d) Is this a BST? Explain your answer.

No, it isn’t. Because it doen’t meet the requirements: the key of left subtree should always be smaller than the key of it’s parent node, the key of right subtree should always be larger than the key of it’s parent node, the left and right subtree of a particular node will also, in turn, be binary search trees.

3. (25 points) Draw the 2-3 tree after inserting each of the keys: 50, 76, 23, 21, 20, 19

(redraw the whole tree for each of the cases). Now, delete key 21 from the tree and redraw the tree. Then, insert key 21 and redraw the tree.

21,50

19,20 23 76

Delete 21:

20,50

19 23 76

Insert 21:

20,50

19 21,23 76