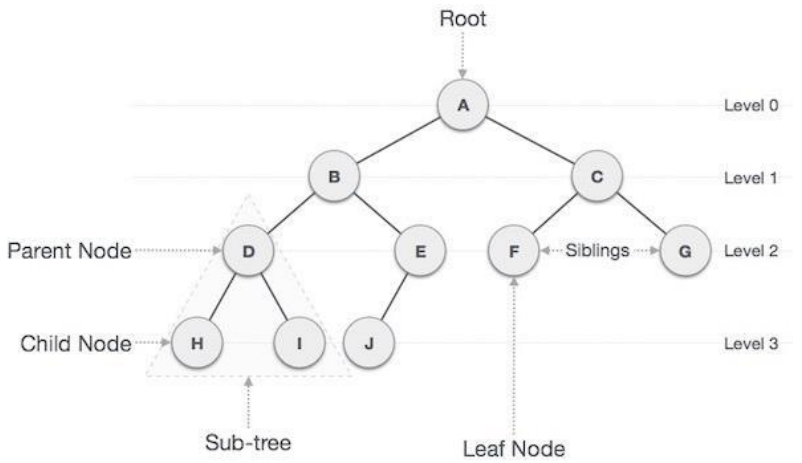
Quiz Section for Program Design (II)

Exercise #9

Being a programmer is exhausting, so let's be a farmer this week. In this exercise, we will implement an abstract data type called "***Binary Search Tree***".

First of all, you should know the basics of a ***Tree***. Table below shows some basic names of tree data type, and the graph below is an example.

Name	Tree
`Node`	Data contained in the tree.
`Edge / Link`	Line connects two nodes.
`Child Node`	For a certain node *N*, the nodes that *N* can go to. A node can have 0 to infinite child nodes.
`Parent Node`	For a certain node *N*, the nodes that can reach back to *N*. A node can only have 1 parent node.
`Root`	The topmost node in the tree, which has no parent node.
`Level`	The distance of a node away from the root.
`Sibling Node`	Nodes in the same level.
`Leaf Node`	Nodes with no child nodes.

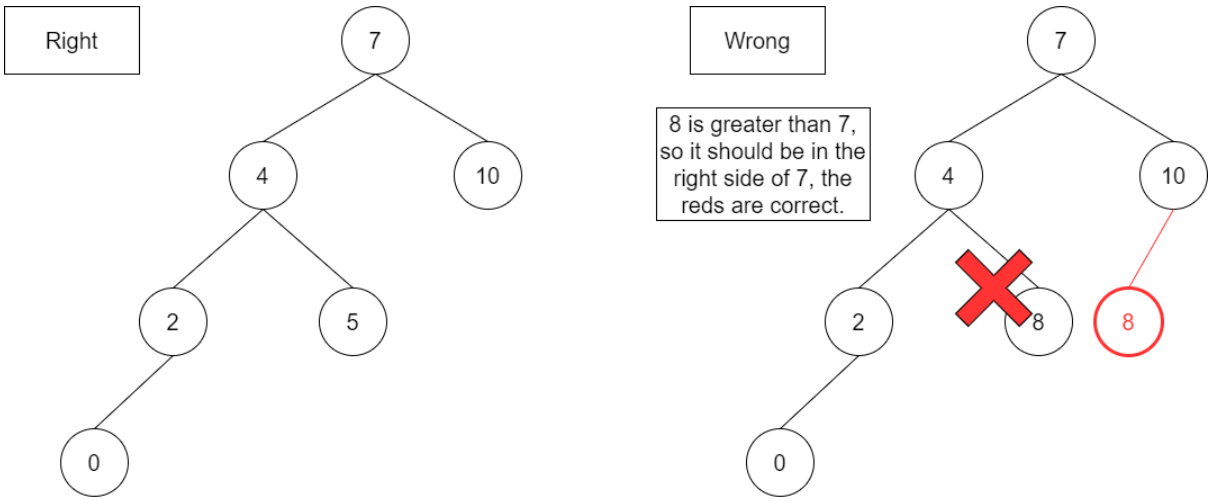


root: A
D's child node: H, I
D's parent node: B
D's sibling node: E, F, G
leaf nodes: H, I, J, F, G

The differences between ***Tree*** and ***Binary Search Tree*** are

1. Every node in the binary search tree can only have utmost 2 child nodes.
2. The value in the left child node must be smaller than its parent node, and the value in the right child node must be greater than the current node (the parent node).

Graph below shows a correct binary search tree and a wrong one.



Your task is to complete the sample code that will be uploaded on eCousrse2. You have to finish the functions below.

function	definition
`Traverse`	Go through the whole tree in the in-order way. In-order traversal is that it will print out all nodes on the left side (which is data smaller than it in this exercise), and then print the current node, so the result should be a sorted array.
`Find`	Find the target node specified by user's input and then return the sum of the data stored in the target node's child nodes. For example, if the target node has two child nodes which store the value 5 and 7. The return value should be 5 plus 7 (i.e., 12). If the target node has no child node, the function needs to return 0. If the function cannot find the target node, it will then return "Not in tree."

The first line of the input will contain one integer, which means the number of data should be inserted into the tree. The second line will be a series of integers

splitted by one space, representing the value of data. It is guaranteed that these data won’t be the same. The inputs in the third line to the end of file will be the targets to find in the tree.

The table below shows the example input and output.

Input	Output
6 7 4 2 5 0 10 2 4 7 11	0 2 4 5 7 10 0 7 14 Not in tree