

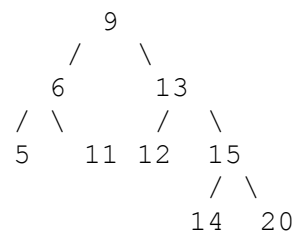
Question 1: Largest Subtree which is a Binary Search Tree (BST)

Given a binary tree, find the largest subtree which is a Binary Search Tree (BST), where largest means subtree with largest number of nodes in it. You should return the root of the largest subtree which is a BST, or null if there is no BST within the given tree. If two or more subtrees are the same size and also the largest BSTs, you may return any of them. You may use the JDK or the standard template library. Your solution will be evaluated on correctness, runtime complexity (big-O), and adherence to coding best practices.

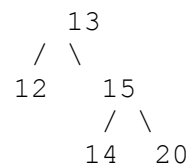
A complete answer will include the following:

1. Document your assumptions
2. Explain your approach and how you intend to solve the problem
3. Provide code comments where applicable
4. Explain the big-O run time complexity of your solution. Justify your answer.

Example:



Answer:



You can use one of the following code skeletons for your solution, or feel free to use the language of your choice.

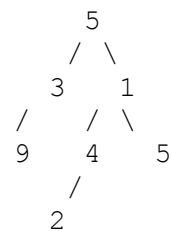
Java: <pre>public class Node { public int value; public Node left; public Node right; } Node findLargestBSTSubtree (Node root) { }</pre>	C++: <pre>typedef struct tnode { int value; struct tnode* left; struct tnode* right; } Node; Node* findLargestBSTSubtree (Node* root) { }</pre>
---	--

Question 2: Tree Printing

You are given a binary tree where each node contains an integer. Print the tree out to the console level by level, with each level on its own line and each node's value separated by a space (but leaving no trailing spaces at the end of a line). You may assume that the tree is non-empty, but not necessarily sorted, balanced, or complete. You may use the JDK or the standard template library. Your solution will be evaluated on correctness, runtime complexity (big-O), and adherence to coding best practices. A complete answer will include the following:

1. Document your assumptions
2. Explain your approach and how you intend to solve the problem
3. Provide code comments where applicable
4. Explain the big-O run time complexity of your solution. Justify your answer.

Example input:



Example output:

```
5
3 1
9 4 5
2
```

You can use one of the following code skeletons for your solution, or feel free to use the language of your choice.

<p>Java:</p> <pre>public class Node { public int value; public Node left; public Node right; } void printTree(Node root) { }</pre>	<p>C++:</p> <pre>typedef struct tnode { int value; struct tnode* left; struct tnode* right; } Node; void printTree (Node* root) { }</pre>
--	---