

Binary Search Tree

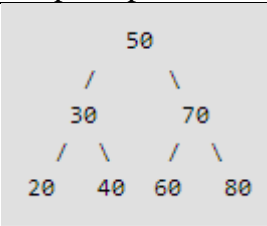
1. Construct a binary search tree from given preorder traversal.

Algorithm:

- i. Take the preorder traversal as input into an array
- ii. Set a global variable idx initialized as 0
- iii. Define a recursive function that does the following. (The function should be called with the parameters –the preorder array, first element of the preorder array, INT_MIN and INT_MAX initially)
 - a. Inside the function, if idx is greater than size of the preorder array, return null otherwise proceed to next step.
 - b. Then check whether the key of the node is in between the MAX and MIN or not. If yes, proceed to the next step.
 - c. Create the new node of the tree. Increment the idx and check idx is greater than the size of the preorder array. If not proceed to the next step. Otherwise return the newly created node.
 - d. Call the function recursively with the preorder array, new value of the updated idx, MIN and the previous value of the key as the MAX.
 - e. Call the function recursively with the preorder array, new value of the updated idx, the previous value of the key as the MIN and the MAX.

| Sample Input | Sample Output |
|---------------------|---|
| 10, 5, 1, 7, 40, 50 | <pre> 10 / \ 5 40 / \ \ 1 7 50</pre> |

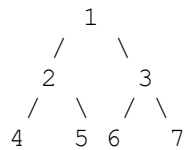
2. Write a program to find kth largest element in BST.

| Sample Input | Sample Output |
|--|---------------|
|  K:2 | Element: 70 |

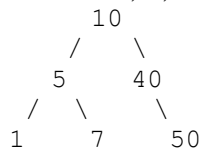
3. Write a program to Remove all internal nodes from the binary search tree

| Sample Input | Sample Output |
|---|--|
| T1 : 50 / \ 20 70 / \ 10 40 | <pre> 50 / \ 40 70 / 10 Or 50 / \ 10 70 \ 40 </pre> |

4. Create a binary tree and determine whether it is a BST or not. For example, you are given some values 1, 2, 3, 4, 5, 6, 7. You have to create a recursive function that creates a BT from the given values. The BT would look like this



This tree is not BST. The output of your program in this case should be false. On the other hand, if the given values are 10, 5, 40, 1, 7, 50 then BT would be like this.



This tree is BST. The output of your program in this case should be true. To solve this problem, after creating BT, run inorder traversal on it. If the traversal is in increasing order then it is surely BST.