Binary Search Tree : Lowest Common Ancestor ★

74 more points to get your next star!

Rank: 565387 | Points: 401/475

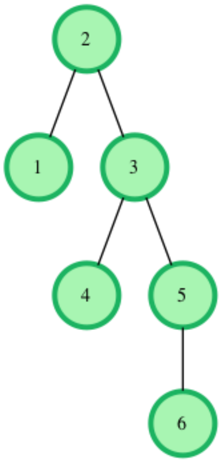
Your Binary Search Tree : Lowest Common Ancestor submission got 30.00 points. Share Post

You are now 74 points away from the 4th star for your problem solving badge.

Try the next challenge | Try a Random Challenge

Problem Submissions Leaderboard Editorial

You are given pointer to the root of the binary search tree and two values $v1$ and $v2$. You need to return the lowest common ancestor (LCA) of $v1$ and $v2$ in the binary search tree.



In the diagram above, the lowest common ancestor of the nodes 4 and 6 is the node 3. Node 3 is the lowest node which has nodes 4 and 6 as descendants.

Function Description

Complete the function lca in the editor below. It should return a pointer to the lowest common ancestor node of the two values given.

lca has the following parameters:

- root: a pointer to the root node of a binary search tree

- v1: a node.data value

- v2: a node.data value

Input Format

The first line contains an integer, n, the number of nodes in the tree.

The second line contains n space-separated integers representing $node.data$ values.

The third line contains two space-separated integers, $v1$ and $v2$.

To use the test data, you will have to create the binary search tree yourself. Here on the platform, the tree will be created for you.

Constraints

$1 \leq n, node.data \leq 25$

$1 \leq v1, v2 \leq 25$

$v1 \neq v2$

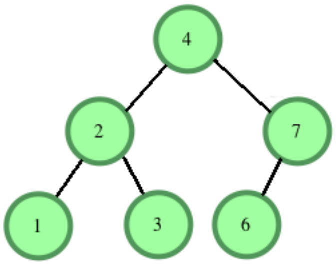
The tree will contain nodes with data equal to $v1$ and $v2$.

Output Format

Return the a pointer to the node that is the lowest common ancestor of $v1$ and $v2$.

Sample Input

```
6
4 2 3 1 7 6
1 7
```



$v1 = 1$ and $v2 = 7$.

Sample Output

[reference to node 4]

Explanation

LCA of 1 and 7 is 4, the root in this case.

Return a pointer to the node.

Change Theme Language Python 3

```
class Node:⋯

# Enter your code here. Read input from STDIN. Print output to STDOUT
'''
class Node:
      def __init__(self,info):
          self.info = info
          self.left = None
          self.right = None


       // this is a node of the tree , which contains info as data, left , right
'''

def lca(root, v1, v2):
    #Enter your code here
    # Ensure v1 is smaller than v2 for easier comparison
    if v1 > v2:
        v1, v2 = v2, v1

    while root:
        # If both values are smaller than current node, go left
        if v2 < root.info:
            root = root.left
        # If both values are larger than current node, go right
        elif v1 > root.info:
            root = root.right
        # Current node is the LCA
        else:
            return root
    return None
tree = BinarySearchTree()⋯
```

EMACS Line: 1 Col: 1

Upload Code as File □ Test against custom input Run Code Submit Code

You have earned 30.00 points!

You are now 74 points away from the 4th star for your problem solving badge.

73% 401/475

Problem Solving ★★★

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Test case 0

Test case 1

Test case 2

Test case 3

Test case 4

Test case 5

Test case 6

Compiler Message

```
Success
```

Input (stdin) Download

```
6
4 2 3 1 7 6
1 7
```

Expected Output Download

```
4
```

Blog | Scoring | Environment | FAQ | About Us | Helpdesk | Careers | Terms Of Service | Privacy Policy