Tree: Height of a Binary Tree

184 more points to get your next star!

Rank: 763584 | Points: 291/475

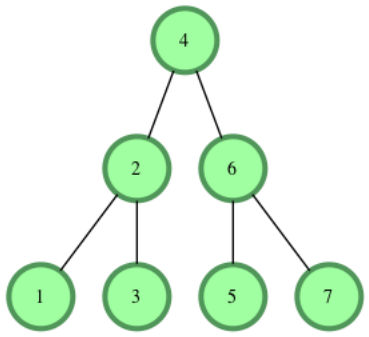
Your Tree: Height of a Binary Tree submission got 10.00 points. Share Post

You are now 184 points away from the 4th star for your problem solving badge.

Try the next challenge | Try a Random Challenge

Problem Submissions Leaderboard Editorial

The height of a binary tree is the number of edges between the tree's root and its furthest leaf. For example, the following binary tree is of height 2:



Function Description

Complete the getHeight or height function in the editor. It must return the height of a binary tree as an integer.

getHeight or height has the following parameter(s):

- root: a reference to the root of a binary tree.

Note -The Height of binary tree with single node is taken as zero.

Input Format

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

Next line contains n space separated integer where ith integer denotes node[i].data.

Note: Node values are inserted into a binary search tree before a reference to the tree's root node is passed to your function. In a binary search tree, all nodes on the left branch of a node are less than the node value. All values on the right branch are greater than the node value.

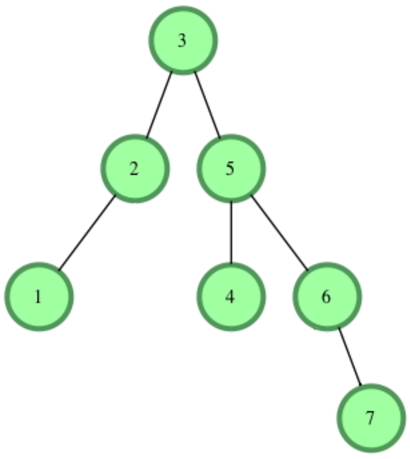
Constraints

$1 \leq node.data[i] \leq 20$
$1 \leq n \leq 20$

Output Format

Your function should return a single integer denoting the height of the binary tree.

Sample Input

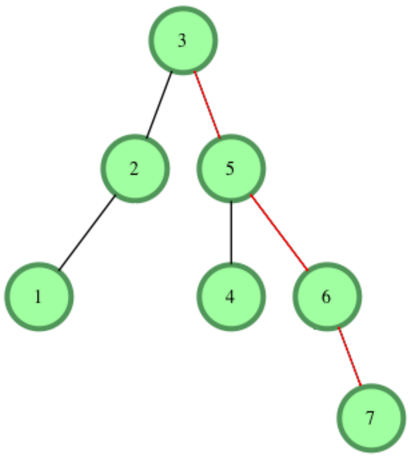


Sample Output

```
3
```

Explanation

The longest root-to-leaf path is shown below:



There are 4 nodes in this path that are connected by 3 edges, meaning our binary tree's $\boldsymbol{height} = 3$.

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 height(root):
    if root is None:
        return -1
    return max(height(root.left), height(root.right)) + 1
⋯
```

EMACS Line: 1 Col: 1

Upload Code as File

Test against custom input

Run Code

Submit Code

You have earned 10.00 points!

You are now 184 points away from the 4th star for your problem solving badge.

33% 291/475

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

Compiler Message

```
Success
```

Input (stdin) Download

```
7
3 5 2 1 4 6 7
```

Expected Output Download

```
3
```