

Data Structures and Algorithms

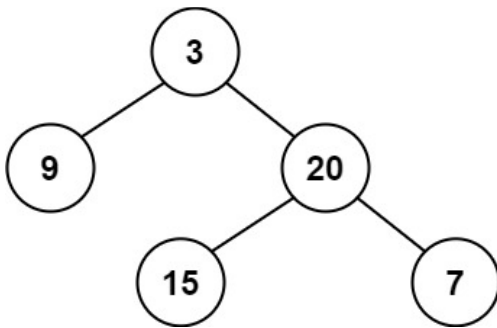
Lab 10 – Binary Search Tree

Exercises/Tasks:

1. Given a binary tree, determine if it is **height-balanced**

A **height-balanced** binary tree is a binary tree in which the depth of the two subtrees of every node **never differs by more than one**.

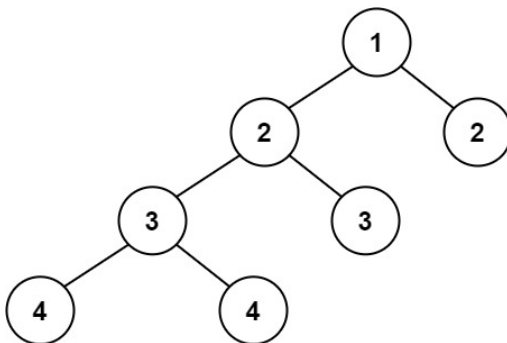
Example 1:



Input: root = [3,9,20,null,null,15,7]

Output: true

Example 2:



Input: root = [1,2,2,3,3,null,null,4,4]

Output: false

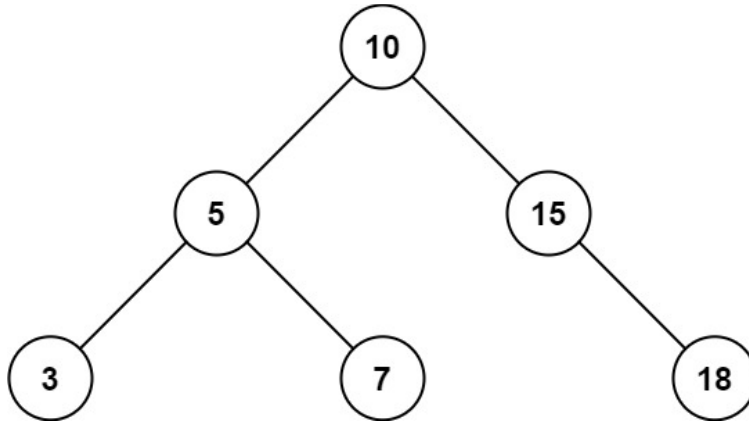
Example 3:

Input: root = []

Output: true

2. Given the root node of a binary search tree and two integers low and high, return *the sum of values of all nodes with a value in the **inclusive** range [low, high]*.

Example 1:

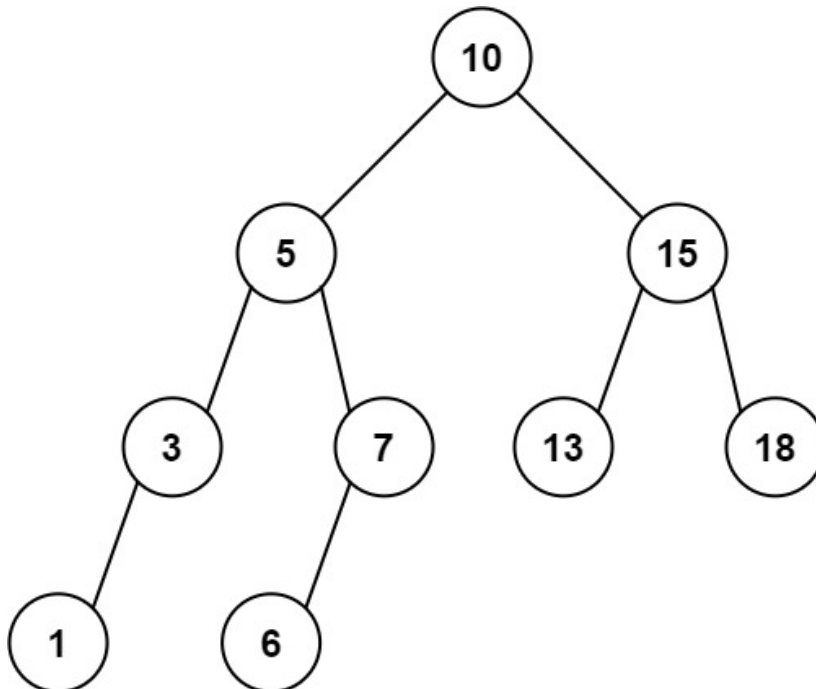


Input: root = [10,5,15,3,7,null,18], low = 7, high = 15

Output: 32

Explanation: Nodes 7, 10, and 15 are in the range [7, 15]. $7 + 10 + 15 = 32$.

Example 2:



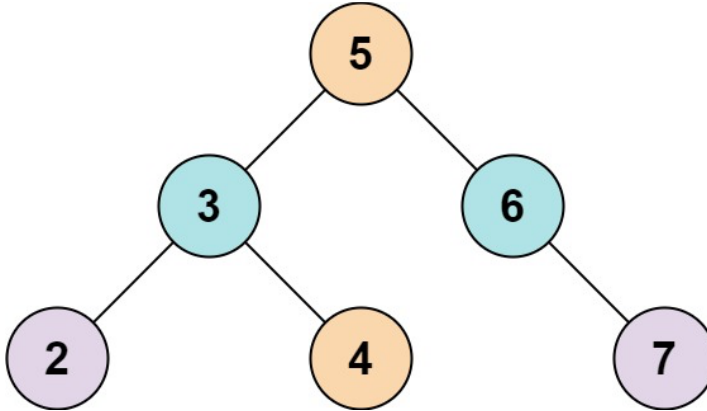
Input: root = [10,5,15,3,7,13,18,1,null,6], low = 6, high = 10

Output: 23

Explanation: Nodes 6, 7, and 10 are in the range [6, 10]. $6 + 7 + 10 = 23$.

3. Given the root of a binary search tree and an integer k , return true *if there exist two elements in the BST such that their sum is equal to k* , or false otherwise.

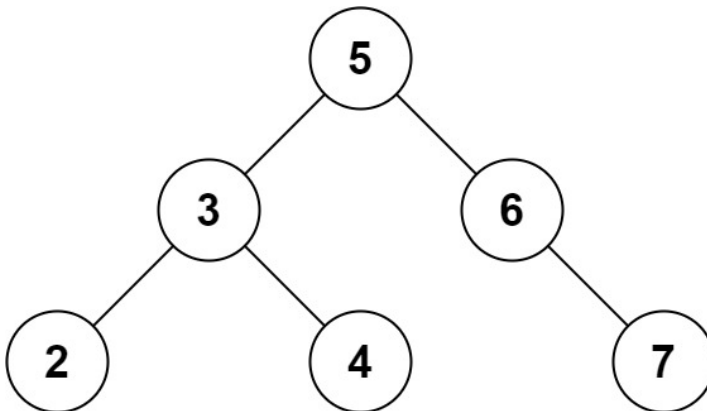
Example 1:



Input: root = [5,3,6,2,4,null,7], $k = 9$

Output: true

Example 2:



Input: root = [5,3,6,2,4,null,7], $k = 28$

Output: false