

Description

Editorial

Solutions (5.9K)

Submissions

112. Path Sum

Easy

8.5K

944

☆

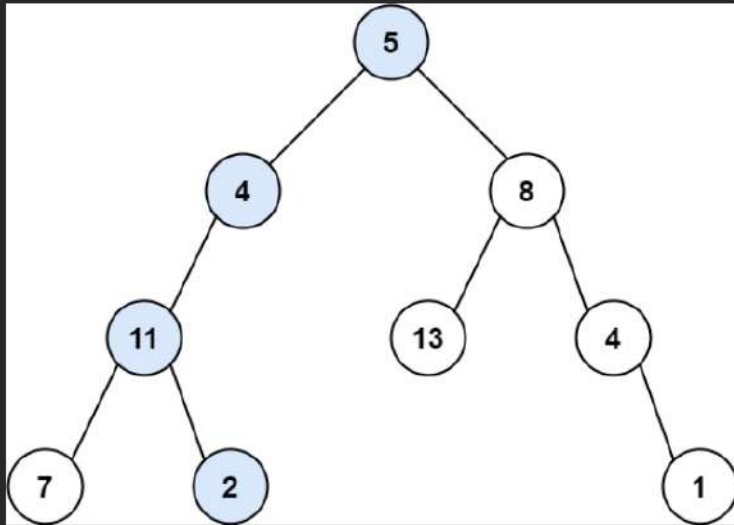
🔄

Companies

Given the `root` of a binary tree and an integer `targetSum`, return `true` if the tree has a **root-to-leaf** path such that adding up all the values along the path equals `targetSum`.

A **leaf** is a node with no children.

Example 1:



Input: `root = [5,4,8,11,null,13,4,7,2,null,null,null,1]`,
`targetSum = 22`

Output: `true`

Explanation: The root-to-leaf path with the target sum is shown.

Java

Auto

```
8     TreeNode(int val) { this.val = val; }
9     TreeNode(int val, TreeNode left, TreeNode right) {
10         this.val = val;
11         this.left = left;
12         this.right = right;
13     }
14 }
15 */
16 class Solution {
17     public boolean hasPathSum(TreeNode root, int targetSum) {
18         if (root == null)
19             return false;
20         if (root.val == targetSum && root.left == null && root.right == null)
21             return true;
22         return hasPathSum(root.left, targetSum - root.val) ||
23             hasPathSum(root.right, targetSum - root.val);
24     }
25 }
```

Testcase

Result

Accepted Runtime: 0 ms

Case 1

Case 2

Case 3

Input

root =

`[5,4,8,11,null,13,4,7,2,null,null,null,1]`

targetSum =

22

Output

true

Expected

Console

❌

Run

Submit

1569. Number of Ways to Reorder Array to Get Same BST

Hint

Hard

1.3K

160

☆

🔄

Companies

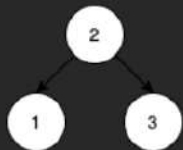
Given an array `nums` that represents a permutation of integers from `1` to `n`. We are going to construct a binary search tree (BST) by inserting the elements of `nums` in order into an initially empty BST. Find the number of different ways to reorder `nums` so that the constructed BST is identical to that formed from the original array `nums`.

- For example, given `nums = [2,1,3]`, we will have 2 as the root, 1 as a left child, and 3 as a right child. The array `[2,3,1]` also yields the same BST but `[3,2,1]` yields a different BST.

Return the number of ways to reorder `nums` such that the BST formed is identical to the original BST formed from `nums`.

Since the answer may be very large, return it modulo $10^9 + 7$.

Example 1:

**Input:** `nums = [2,1,3]`**Output:** 1**Explanation:** We can reorder `nums` to be `[2,3,1]` which will yield the same BST. There are no other ways to reorder `nums` which will yield the same BST.

Example 2:



Java Auto

```
1 class Solution {
2     public int numOfWays(int[] nums) {
3         List<Integer> arr=new ArrayList<>();
4         for(int i=0; i<nums.length; i++) {
5             arr.add(nums[i]);
6         }
7
8         int n=nums.length;
9         pascal=new long[n+1][n+1];
10        pascal[0][0]=1;
11        //nC0 = n-1C0-1 + n-1C0
12        for(int i=1; i<=n; i++) {
13            pascal[i][0]=1;
14            for(int j=1; j<=i; j++) {
15                pascal[i][j] = (pascal[i-1][j-1]+pascal[i-1][j])%MOD;
16            }
17            pascal[i][i]=1;
18        }
19    }
20 }
```

Testcase Result

Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3

Input

`nums =``[2,1,3]`

Output

1

Expected

1

Console



Run

Submit