Problem 2196: Create Binary Tree From Descriptions

Problem Information

Difficulty: Medium
Acceptance Rate: 81.66%
Paid Only: No
Tags: Array, Hash Table, Tree, Binary Tree

Problem Description

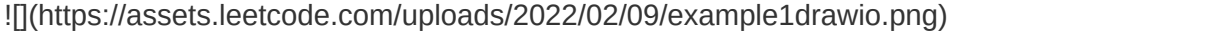
You are given a 2D integer array `descriptions` where `descriptions[i] = [parenti, childi, isLefti]` indicates that `parenti` is the **parent** of `childi` in a **binary** tree of **unique** values. Furthermore,

* If `isLefti == 1`, then `childi` is the left child of `parenti`. * If `isLefti == 0`, then `childi` is the right child of `parenti`.

Construct the binary tree described by `descriptions` and return _its**root**_.

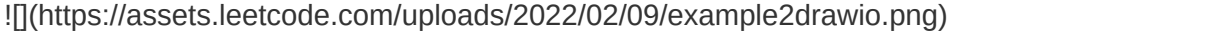
The test cases will be generated such that the binary tree is **valid**.

Example 1:



Input: descriptions = [[20,15,1],[20,17,0],[50,20,1],[50,80,0],[80,19,1]] **Output:** [50,20,80,15,17,19] **Explanation:** The root node is the node with value 50 since it has no parent. The resulting binary tree is shown in the diagram.

Example 2:



Input: descriptions = [[1,2,1],[2,3,0],[3,4,1]] **Output:** [1,2,null,null,3,4] **Explanation:** The root node is the node with value 1 since it has no parent. The resulting binary tree is shown in the diagram.

Constraints:

* `1 <= descriptions.length <= 104` * `descriptions[i].length == 3` * `1 <= parenti, childi <= 105` * `0 <= isLefti <= 1` * The binary tree described by `descriptions` is valid.

Code Snippets

C++:

```
/**
* Definition for a binary tree node.
* struct TreeNode {
* int val;
* TreeNode *left;
* TreeNode *right;
* TreeNode() : val(0), left(nullptr), right(nullptr) {}
* TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
* TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left),
right(right) {}
* };
*/
class Solution {
public:
TreeNode* createBinaryTree(vector<vector<int>>& descriptions) {

}
};
```

Java:

```
/**
* Definition for a binary tree node.
* public class TreeNode {
* int val;
* TreeNode left;
* TreeNode right;
* TreeNode() {}
```

```
* TreeNode(int val) { this.val = val; }
* TreeNode(int val, TreeNode left, TreeNode right) {
* this.val = val;
* this.left = left;
* this.right = right;
* }
* }
*/
class Solution {
public TreeNode createBinaryTree(int[][] descriptions) {

}
}
```

Python3:

```
# Definition for a binary tree node.
# class TreeNode:
# def __init__(self, val=0, left=None, right=None):
# self.val = val
# self.left = left
# self.right = right
class Solution:
def createBinaryTree(self, descriptions: List[List[int]]) ->
Optional[TreeNode]:
```