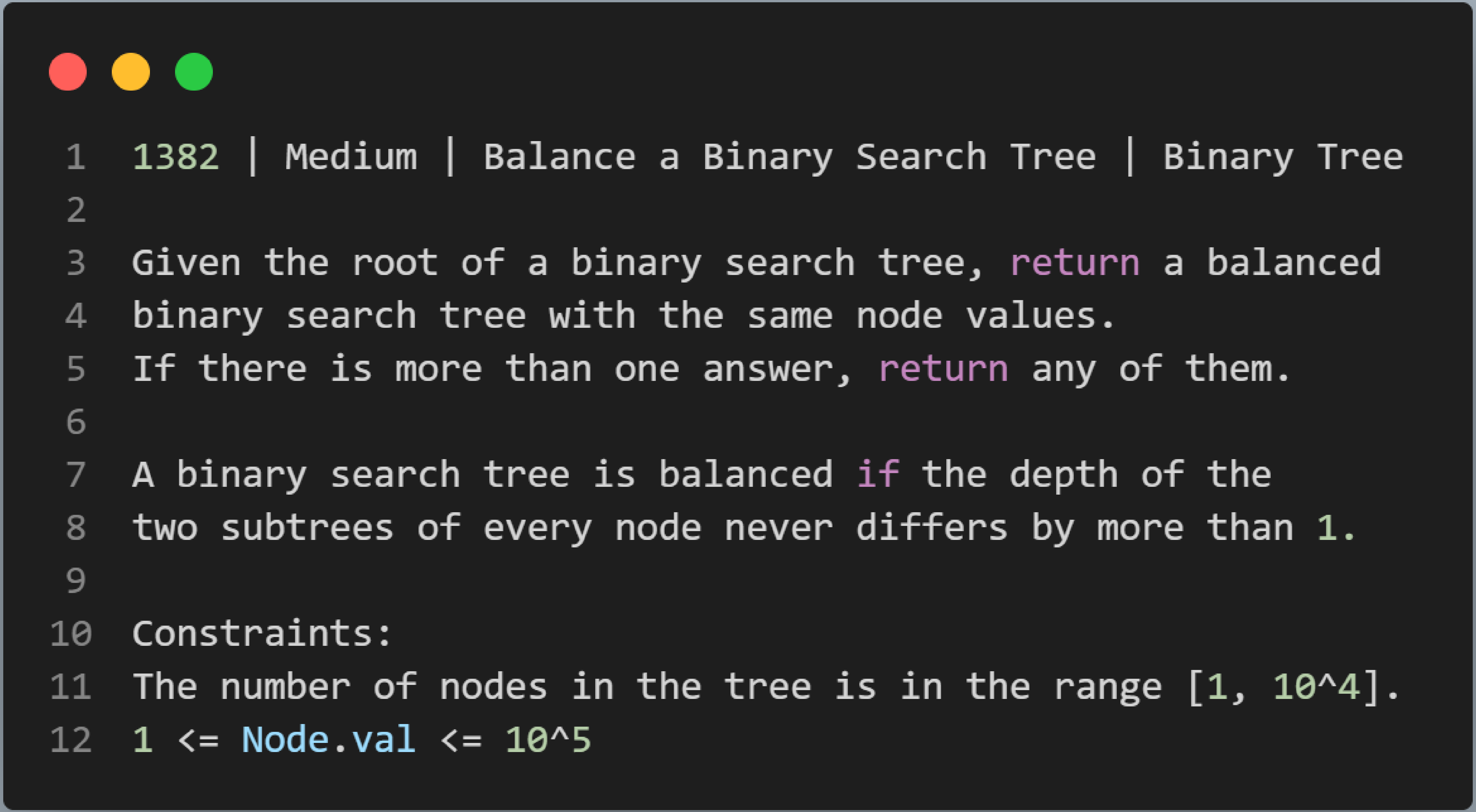
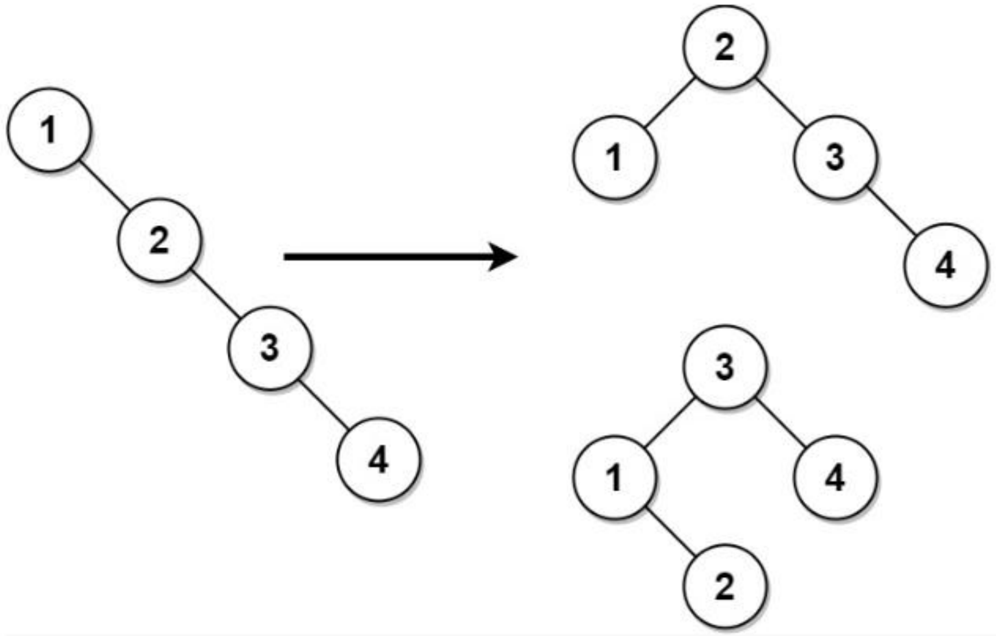


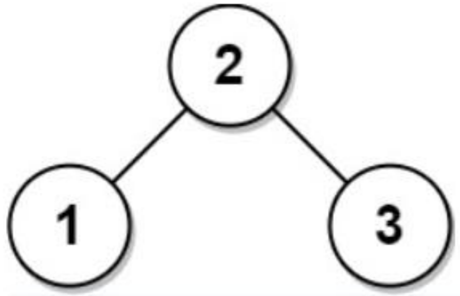
```
1382 | Medium | Balance a Binary Search Tree | Binary Tree

Given the root of a binary search tree, return a balanced
binary search tree with the same node values.
If there is more than one answer, return any of them.

A binary search tree is balanced if the depth of the
two subtrees of every node never differs by more than 1.

Constraints:
The number of nodes in the tree is in the range [1, 10^4].
1 <= Node.val <= 10^5
```

Example 1:

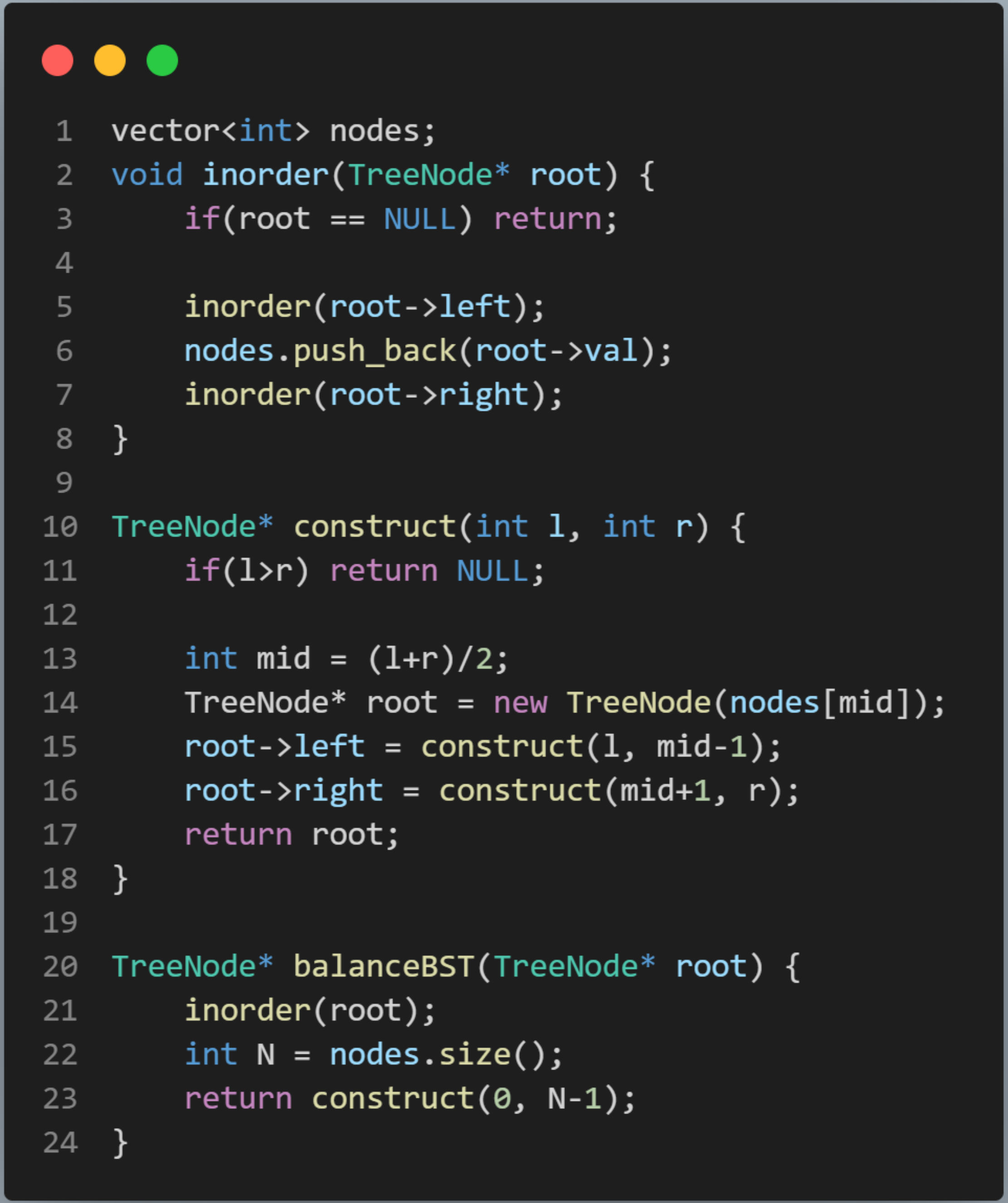


```
Input: root = [1,null,2,null,3,null,4,null,null]
Output: [2,1,3,null,null,null,4]
Explanation: This is not the only correct answer, [3,1,4,null,2] is also correct.
```

Example 2:



```
Input: root = [2,1,3]
Output: [2,1,3]
```



```cpp
vector<int> nodes;
void inorder(TreeNode* root) {
    if(root == NULL) return;

    inorder(root->left);
    nodes.push_back(root->val);
    inorder(root->right);
}

TreeNode* construct(int l, int r) {
    if(l>r) return NULL;

    int mid = (l+r)/2;
    TreeNode* root = new TreeNode(nodes[mid]);
    root->left = construct(l, mid-1);
    root->right = construct(mid+1, r);
    return root;
}

TreeNode* balanceBST(TreeNode* root) {
    inorder(root);
    int N = nodes.size();
    return construct(0, N-1);
}
```

#100daysofDSA

/rvislive

Rakesh Vishwakarma