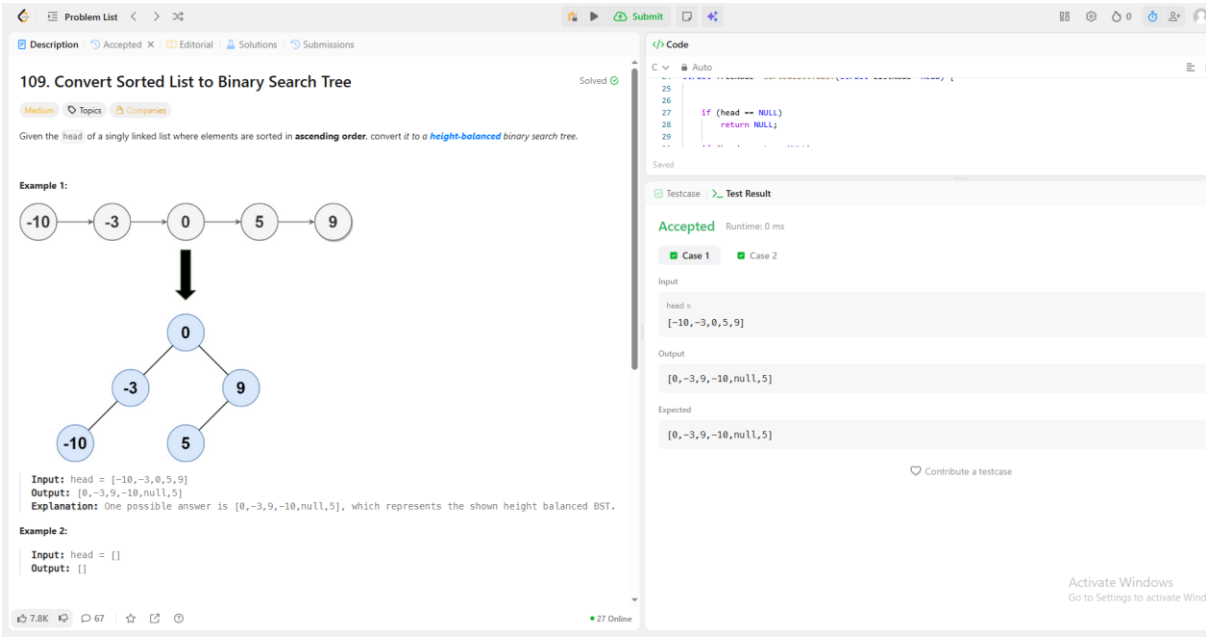
Convert Sorted List to Binary Search Tree

```c
struct TreeNode* createNode(int val) {
    struct TreeNode* node = (struct TreeNode*)malloc(sizeof(struct TreeNode));
    node->val = val;
    node->left = NULL;
    node->right = NULL;
    return node;
}

struct TreeNode* sortedListToBST(struct ListNode* head) {


    if (head == NULL)
        return NULL;

    if (head->next == NULL)
        return createNode(head->val);


    struct ListNode *slow = head, *fast = head, *prev = NULL;

    while (fast != NULL && fast->next != NULL) {
        prev = slow;
        slow = slow->next;
        fast = fast->next->next;
    }


    prev->next = NULL;


    struct TreeNode* root = createNode(slow->val);

    root->left = sortedListToBST(head);
    root->right = sortedListToBST(slow->next);

    return root;
}
```

Output


Problem List
Submit
Description
Accepted
Editorial
Solutions
Submissions
109. Convert Sorted List to Binary Search Tree
Solved
Medium
Topics
Companies
Given the head of a singly linked list where elements are sorted in ascending order, convert it to a height-balanced binary search tree.
Example 1:
-10
-3
0
5
9
Input: head = [-10,-3,0,5,9]
Output: [0,-3,9,-10,null,5]
Explanation: One possible answer is [0,-3,9,-10,null,5], which represents the shown height balanced BST.
Example 2:
Input: head = []
Output: []
7.8K
67
27 Online
Code
Auto
if (head == NULL)
return NULL;
Saved
Testcase
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Input
head =
[-10,-3,0,5,9]
Output
[0,-3,9,-10,null,5]
Expected
[0,-3,9,-10,null,5]
Contribute a testcase
Activate Windows