

## Leetcode 109:

Problem List < > 109

Description Editorial Solutions Submissions

### 109. Convert Sorted List to Binary Search Tree

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:**

**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:** []

**Constraints:**

786 67 23 Online

## Solution:

Problem List < > 109

Code

C++ Auto

```
1 //
2 // Definition for singly-linked list.
3 struct ListNode {
4     int val;
5     struct ListNode *next;
6 };
7 //
8 // Definition for a binary tree node.
9 struct TreeNode {
10     int val;
11     struct TreeNode *left;
12     struct TreeNode *right;
13 };
14 //
15 //
16 struct TreeNode* sortedListToBST(struct ListNode* head) {
17     if (head == NULL)
18         return NULL;
19
20     if (head->next == NULL) {
21         struct TreeNode* node = (struct TreeNode*)malloc(sizeof(struct TreeNode));
22         node->val = head->val;
23         node->left = NULL;
24         node->right = NULL;
25         return node;
26     }
27
28     struct ListNode *slow = head, *fast = head, *prev = NULL;
29
30     while (fast && fast->next) {
31         prev = slow;
32         slow = slow->next;
33         fast = fast->next->next;
34     }
35
36     prev->next = NULL;
37
38     struct TreeNode* root = (struct TreeNode*)malloc(sizeof(struct TreeNode));
39     root->val = slow->val;
40
41     root->left = sortedListToBST(head);
42     root->right = sortedListToBST(slow->next);
43
44     return root;
45 }
```

Solved

Ln 1, Col 1

## Test Case:

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](#)