109. Convert Sorted List to Binary Search Tree
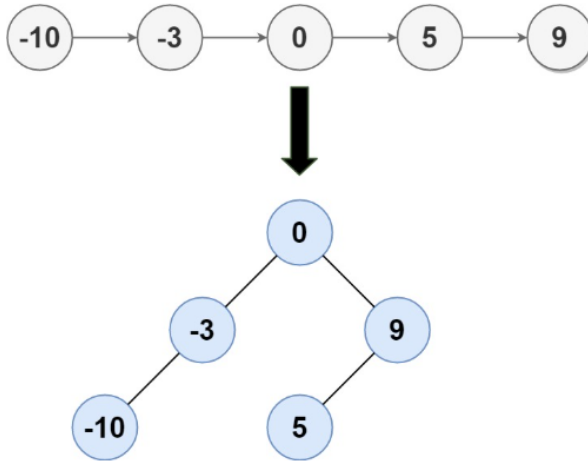
Medium | Topics | Companies

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: []
```

```c
struct TreeNode* sortedListToBST(struct ListNode* head) {
    if (head == NULL)
    return NULL;

/* If only one node, it becomes the root */
if (head->next == NULL) {
    struct TreeNode* root = (struct TreeNode*)malloc(sizeof(struct TreeNode));
    root->val = head->val;
    root->left = NULL;
    root->right = NULL;
    return root;
}

/* Find middle of linked list */
struct ListNode *slow = head, *fast = head, *prev = NULL;

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

/* Break the list into two halves */
prev->next = NULL;

/* Middle element becomes root */
struct TreeNode* root = (struct TreeNode*)malloc(sizeof(struct TreeNode));
root->val = slow->val;

/* Recursively build left and right subtrees */
root->left = sortedListToBST(head);
root->right = sortedListToBST(slow->next);

return root;

}
```

☑ 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

☑ Testcase | >_ **Test Result**

Accepted Runtime: 0 ms

☑ Case 1   ☑ **Case 2**

Input

```
head =
[]
```

Output

```
[]
```

Expected

```
[]
```

♡ Contribute a testcase