

## 8b. Leetcode 109

### Convert Sorted List to Binary Search Tree

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
// Definition for singly-linked list
```

```
// Helper function to find middle node
```

```
struct ListNode* findMiddle(struct ListNode* head, struct ListNode** prev) {
```

```
    struct ListNode *slow = head;
```

```
    struct ListNode *fast = head;
```

```
    *prev = NULL;
```

```
    while (fast != NULL && fast->next != NULL) {
```

```
        *prev = slow;
```

```
        slow = slow->next;
```

```
        fast = fast->next->next;
```

```
    }
```

```
    return slow;
```

```
}
```

```
// Main function to convert sorted list to BST
```

```
struct TreeNode* sortedListToBST(struct ListNode* head) {
```

```
    if (head == NULL)
```

```
        return NULL;
```

```
__ struct ListNode *prev = NULL;
__ struct ListNode *mid = findMiddle(head, &prev);

__ // Create tree node for middle element
__ struct TreeNode *root = (struct TreeNode*)malloc(sizeof(struct TreeNode));
__ root->val = mid->val;
__ root->left = root->right = NULL;

__ // If mid is head, left subtree is NULL
__ if (prev != NULL) {
__     prev->next = NULL; // Disconnect left half
__     root->left = sortedListToBST(head);
__ }

__ root->right = sortedListToBST(mid->next);

__ return root;
__ }
```

OUTPUT:

**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]
```

---

✓ Case 1

✓ Case 2

Input

```
head =  
[]
```

Output

```
[]
```

Expected

```
[]
```

---