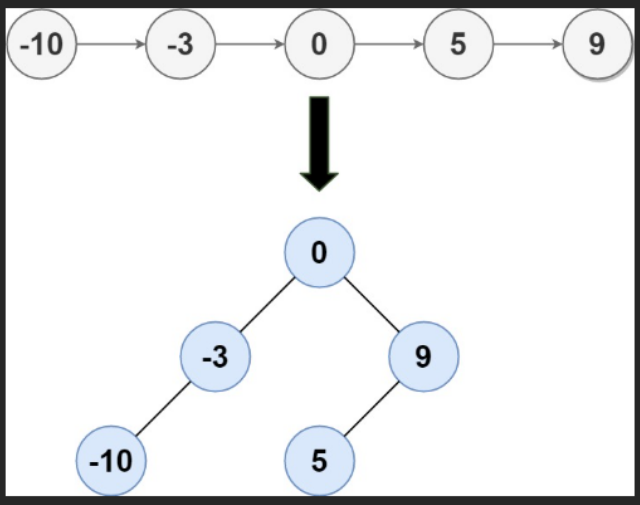
LEETCODE PROBLEM – 109

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.
```

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]
```

Code

C Auto

```c
struct TreeNode {
    int val;
    struct TreeNode *left;
    struct TreeNode *right;
};

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

struct ListNode* getMiddle(struct ListNode* head) {
    struct ListNode *slow = head, *fast = head, *prev = NULL;

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

    if (prev != NULL)
        prev->next = NULL;

    return slow;
}

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

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

    struct ListNode* mid = getMiddle(head);
    struct TreeNode* root = newNode(mid->val);

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

    return root;
}

```