

Code-

```
class Solution {
public:
    TreeNode* sortedListToBST(ListNode* head) {
        if (!head) {
            return nullptr;
        }
        ListNode* mid = findMiddle(head);
        TreeNode* root = new TreeNode(mid->val);
        if (head == mid) {
            return root;
        }
        root->left = sortedListToBST(head);
        root->right = sortedListToBST(mid->next);
        return root;
    }

    ListNode* findMiddle(ListNode* head) {
        ListNode* slow = head;
        ListNode* fast = head;
        ListNode* prev = nullptr;
        while (fast && fast->next) {
            prev = slow;
            slow = slow->next;
            fast = fast->next->next;
        }
        if (prev) {
            prev->next = nullptr;
        }
        return slow;
    }
};
```

Time Complexity-  $O(N)$

Space Complexity- $O(N)$

