

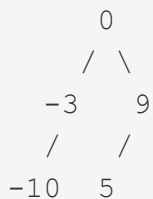
给定一个单链表，其中的元素按升序排序，将其转换为高度平衡的二叉搜索树。

本题中，一个高度平衡二叉树是指一个二叉树每个节点的左右两个子树的高度差的绝对值不超过 1。

### 示例:

给定的有序链表: [-10, -3, 0, 5, 9],

一个可能的答案是: [0, -3, 9, -10, null, 5], 它可以表示下面这个高度平衡二叉搜索树:



思路，平衡二叉树，就是数从升序的中间把树劈开，分为左右两半，左边指小的，右边指大的，按照这个思路，进行树的递归，即可得到答案，需注意，当节点为1时，需避免重复的递归，直接返回该节点。

### 代码

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode(int x) : val(x), next(NULL) {}
 * };
 */
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode(int x) : val(x), left(NULL), right(NULL) {}
 * };
 */
class Solution {
public:
    int listlength(ListNode* head){
        if(head==NULL){
            return 0;
        }
        return 1+listlength(head->next);
    }
}
```

```

TreeNode* dfs(ListNode* head){
    if(head==NULL){
        return NULL;
    }
    int length=listlength(head);
    ListNode* right=head;
    int num=1;
    while(num<=length/2){
        num++;
        ListNode* last=right;
        right=right->next;
        if(num>length/2){
            last->next=NULL;
        }
    }
    TreeNode *root=new TreeNode(right->val);
    if(length==1){
        return root;
    }
    root->left=dfs(head);
    root->right=dfs(right->next);
    return root;
}
TreeNode* sortedListToBST(ListNode* head) {
    return dfs(head);
}
};

```