109. Convert Sorted List to Binary Search Tree

题目描述：https://leetcode.com/problems/convert-sorted-list-to-binary-search-tree/

> 给定一个链表，要求把它转化成为高度最高的平衡二叉树
> 例如：[1,2,3,4,5,6,7,8,9]
>
> ```
>         5
>       /   \
>      3     8
>     / \   / \
>    2   4 7   9
>   /     /
> 1      6
> ```

解题思路：

找到根节点其实就是中间的mid节点，然后递归生成左边子树和右边子树

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:
    TreeNode* sortedListToBST(ListNode* head) {
        if(head == NULL)
            return NULL;
        if(head->next == NULL){
            TreeNode *root = new TreeNode (head->val);
            return root;
        }
        ListNode* mid = head, *last = head->next, *pre;
        while(last && last->next){
            pre = mid;
            mid = mid->next;
            last = last->next->next;
        }
        if(last != NULL){
            pre = mid;
            mid = mid->next;
        }
        pre->next = NULL;
        TreeNode * root = new TreeNode(mid->val);
        cout << "root:" << root->val<<endl;
        root->right = sortedListToBST(mid->next);
        root->left = sortedListToBST(head);
        return root;
    }
};
```