```cpp
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode(int x) : val(x), next(NULL) {}
 * };
 */
class Solution {
public:
    ListNode* detectCycle(ListNode* head) {
        ListNode* current = head;
        //如果头节点为空，直接返回空指针
        if (current == nullptr) {
            return nullptr;
        }
        //定义哈希表
        unordered_map<ListNode*, int> hashtable;
        int i = 0;
        while (1) {
            auto it = hashtable.find(current);
            //碰上环了
            if (it != hashtable.end()) {
                return it->first;
                break;
            }
            //还没环上
            else {
                hashtable[current] = i;
                i++;
                current = current->next;
            }
            //对于没有环的，保证要保证current不为空，
            //也就是遍历完毕链表了，还没有找到环。
            if (current == nullptr) {
                return nullptr;
                break;
            }
        }
    }
};```
```

题解

```cpp
class Solution {
public:
    ListNode *detectCycle(ListNode *head) {
        unordered_set<ListNode *> visited;
        while (head != nullptr) {
            if (visited.count(head)) {
                return head;
            }
            visited.insert(head);
            head = head->next;
        }
        return nullptr;
    }
};
```