一、题目说明

题目142. Linked List Cycle II，判断一个链表是否有环，如果有返回环的第一个元素，否则返回NULL。

这个题目是**141. Linked List Cycle**的升级版本，难度是Medium！

二、我的解答

最直观的解答就是用一个unordered_map<ListNode*,int> dp来统计节点出现的次数，如果出现2，则这个就是第一个节点。

```
class Solution{
    public:
        ListNode* detectCycle(ListNode* head){
            if(head==NULL || head->next==NULL){
                return NULL;
            }
            unordered_map<ListNode*,int> dp;
            dp[head] = 1;
            while(head!=NULL && dp[head]<2){
                head = head->next;
                if(dp.count(head)>0){
                    dp[head]++;
                }
            }
            return head;
        }
};
```

性能：

```
Runtime: 20 ms, faster than 24.70% of C++ online submissions for Linked List
Cycle II.
Memory Usage: 12.5 MB, less than 7.14% of C++ online submissions for Linked List
Cycle II.
```

三、优化措施

不使用额外的空间，就要用上一个题目**141. Linked List Cycle**的fast，slow双指针法了。

```
class Solution{
    public:
        ListNode* detectCycle(ListNode* head){
            if(head==NULL || head->next==NULL){
                return NULL;
            }
            ListNode* fast=head,*slow=head;
            while(fast && fast->next){
                fast= fast->next->next;
                slow = slow->next;

                if(fast == slow){
                    slow = head;
```

```
                    while(fast!=slow){
                        fast = fast->next;
                        slow = slow->next;
                    }
                    return fast;
                }
            }

            return NULL;
        }
};
```

性能如下：

```
Runtime: 4 ms, faster than 99.87% of C++ online submissions for Linked List
Cycle II.
Memory Usage: 9.9 MB, less than 59.52% of C++ online submissions for Linked List
Cycle II.
```