142 环形链表II

Label：双指针
给定一个链表，返回链表开始入环的第一个节点。 如果链表无环，则返回 null。

- 双指针 可以推导出公式

```
public class Solution {
    public ListNode detectCycle(ListNode head) {
         if (head == null || head.next == null) return null;
        // 步骤一：使用快慢指针判断链表是否有环
        ListNode p = head, p2 = head;
        boolean hasCycle = false;
        while (p2.next != null && p2.next.next != null) {
            p = p.next;
            p2 = p2.next.next;
            if (p == p2) {
                hasCycle = true;
                break;
            }
        }
        // 步骤二：若有环，找到入环开始的节点
        if (hasCycle) {
            ListNode q = head;
            while (p != q) {
                p = p.next;
                q = q.next;
            }
            return q;
        } else
            return null;  // 无环，返回null
        }
}
```

- Hash==>判断重复，第一想到的肯定是Hash

```
public class Solution {
    public ListNode detectCycle(ListNode head) {
        Set<ListNode> set = new HashSet<>();
        while (head != null) {
            if (set.contains(head)) {
                return head;
            }else {
                set.add(head);
            }
            head = head.next;
        }
        return null;
    }
}
```