1.数组去重问题

Hint: 首先将数组排序，这样那些重复的整数就会被挤在一起。然后用两根指针，一根指针走得快一些遍历整个数组，另外一根指针，一直指向当前不重复部分的最后一个数。快指针发现一个和慢指针指向的数不同的数之后，就可以把这个数丢到慢指针的后面一个位置，并把慢指针++。

Practice:

- Move Zeros
- Two Sum III — Data Structure Design
- Remove Duplicate Numbers in Array
- Two Sum — Difference equals to target

```
public class Solution {
    /**
     * @param nums: an array of Integer
     * @param target: an integer
     * @return: [index1 + 1, index2 + 1] (index1 < index2)
     */
    public int[] twoSum7(int[] nums, int target) {
        HashMap<Integer, Integer> map = new HashMap<>();
        for (int i = 0; i < nums.length; i++) {
            int sum = nums[i] + target;
            if (map.containsKey(sum)) {
                int index = map.get(sum);
                int[] pair = new int[2];
                pair[0] = index + 1;
                pair[1] = i + 1;
                return pair;
            }

            int diff = nums[i] - target;
            if (map.containsKey(diff)) {
                int index = map.get(diff);
                int[] pair = new int[2];
                pair[0] = index + 1;
                pair[1] = i + 1;
                return pair;
            }
            map.put(nums[i], i);
        }

        return null;
    }
}
```

2.滑动窗口问题

Sliding window

在专门Sliding Window的note中说明。

3.链表中点问题

Middle of LinkedList

Hint: 1.快慢同向指针实现

2.需要先判断特殊情况（长度<= 3的情况）

```
public ListNode middleNode(ListNode head) {
    // write your code here
    if(head == null){
        return null;
    }
    if(head.next == null){
        return head;
    }
    ListNode slow = head;
    ListNode fast = slow.next;
    if(fast.next == null){
        return slow;
    }
    if(fast.next.next == null){
        return fast;
    }
    while(fast.next != null && fast.next.next != null){
        slow = slow.next;
        fast = fast.next.next;
    }
    if(fast.next == null){
        return slow;
    }
    else{
        return slow.next;
    }
}
```