

Hot100 128、最长连续序列（M）

题目描述：给定一个未排序的整数数组 `nums`，找出数字连续的最长序列（不要求序列元素在原数组中连续）的长度。

要求O(N)

输入：`nums = [10,4,20,1,3,2]`

输出：4

评价，这题很赞，考察区间处理和【复杂度的理解】，也可以用并查集思路

思路，先排序的话就很容易了，但是会达到 $n\lg n$ 复杂度

既然要ON，就需要额外处理哈希结构

1、用哈希整理一下数组，以备快速查询数N，的N-1和N+1是否存在O（1）

2、遍历数组，处理每个N

如果N-1存在，则跳过（因为它必定在一个序列之内）；

如果N-1不存在说明是开头，那就循环看N+1；

虽然这里有两重循环，但是对于每一个N，只会进入其中一个循环，所以O（N）

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

首先用一个哈希表set存起来nums，然后我们看如何遍历

10	4	20	1	3	2		查询9，不存在，开始判断11，不存在											
10	4	20	1	3	2		查询3，存在，跳过，4一定不是起始节点											
10	4	20	1	3	2		查询19，不存在，查询21，不存在											
10	4	20	1	3	2		查询0，不存在，说明1一定是起点；											
10	4	20	1	3	2		查询2，跳过											
10	4	20	1	3	2		查询1，跳过											

```
public int longestConsecutive(int[] nums) {  
    Set<Integer> set = new HashSet<>();  
    for (int i : nums){  
        set.add(i);  
    }  
    int max = 0;  
    for(int i : set){  
        if(set.contains(i-1)){//如果存在n-1, 说明n一定不是头, 直接跳过  
            continue;  
        }  
        int count = 1;  
        int step = 1;  
        while (set.contains(i+step)){//如果不存在, 循环看后续是否存在  
            count ++;  
            step ++;  
        }  
        max = Math.max(max, count);  
    }  
    return max;  
}
```