

349. 两个数组的交集 + 202. 快乐数

Friday, July 15, 2022 9:29 AM

<https://leetcode.cn/problems/intersection-of-two-arrays/>

简单

给定两个数组 nums1 和 nums2，返回 它们的交集。输出结果中的每个元素一定是 唯一的。

我们可以 不考虑输出结果的顺序。

示例 1:

输入: nums1 = [1,2,2,1], nums2 = [2,2]

输出: [2]

示例 2:

输入: nums1 = [4,9,5], nums2 = [9,4,9,8,4]

输出: [9,4]

解释: [4,9] 也是可通过的

思路: set过滤

```
class Solution {
    public int[] intersection(int[] nums1, int[] nums2) {
        if(nums1.length==0||nums2.length==0){
            return new int[]{};
        }
        HashSet<Integer> set=new HashSet<>(); //过滤nums1
        HashSet<Integer> set2=new HashSet<>(); //nums2中也有的放入过滤
        for(int i:nums1){
            set.add(i);
        }
        for(int i:nums2){
            if(set.contains(i)){
                set2.add(i);
            }
        }
        int[] res=new int[set2.size()]; // 结果数组
        int index=0;
        for(int i:set2){
            res[index++]=i;
        }
        return res;
    }
}
```

202. 快乐数

<https://leetcode.cn/problems/happy-number/>

简单

编写一个算法来判断一个数 n 是不是快乐数。

「快乐数」定义为：

对于一个正整数，每一次将该数替换为它每个位置上的数字的平方和。

然后重复这个过程直到这个数变为 1，也可能是无限循环但始终变不到 1。

如果这个过程结果为 1，那么这个数就是快乐数。

如果 n 是快乐数就返回 true；不是，则返回 false。

示例 1：

输入： $n = 19$

输出：true

解释：

$1^2 + 9^2 = 82$

$8^2 + 2^2 = 68$

$6^2 + 8^2 = 100$

$1^2 + 0^2 + 0^2 = 1$

示例 2：

输入： $n = 2$

输出：false

思路：用set判断中间过程有没有出现重复结果，要是有则肯定死循环了。

```
class Solution {
    public boolean isHappy(int n) {
        Set<Integer> record = new HashSet<>(); //存中间结果
        while (n != 1 && !record.contains(n)) { // n不为1且n没出现过 进入循环
            record.add(n); // 记录n
            n = getNextNumber(n); // 得到此时n拆开的平方和结果
        }
        return n == 1; //最终返回 n是否为1
    }

    private int getNextNumber(int n) { // 计算中间平方和结果
        int res = 0;
        while (n > 0) {
            int temp = n % 10;
            res += temp * temp;
            n = n / 10;
        }
        return res;
    }
}
```

}
}