# 查找

## LeetCode 第 350 题：

**题目描述**

给定两个数组 nums,求两个数组的交集。

如 nums1=[1,2,2,1],nums=[2,2
结果为[2,2]
出现的顺序可以是任意的。

代码：

```
class Solution:
····def intersect(self, nums1: List[int], nums2: List[int]) -> List[int]:
·········from collections import Counter
         nums1_dict = Counter(nums1)
         res = []
         for num in nums2:
         if nums1_dict[num] > 0:
         res.append(num)
         nums1_dict[num] -= 1
         return res
```

## LeetCode 第 242 题：

**题目描述**

给定两个字符串 *s* 和 *t* ，编写一个函数来判断 *t* 是否是 *s* 的字母异位词。

示例 1:
输入: s = "anagram", t = "nagaram"
输出: true
示例 2:
输入: s = "rat", t = "car"
输出: false

代码如下：

```
class Solution:
def isAnagram(self, s: str, t: str) -> bool:
        from collections import Counter
        d1=Counter(s)
        d2=Counter(t)
        if d1==d2:
            return True
```

```
        else:
            return False
```

## LeetCode 第 202 题：

**题目描述**

编写一个算法来判断一个数是不是“快乐数”。 一个“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。如果可以变为 1，那么这个数就是快乐数。

示例:
输入: 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

代码如下：

```
class Solution:
def isHappy(self, n: int) -> bool:
    s=set()
    S.add(n)
    While n!=1:
        C=0
        For i in str(s):
            C+=int(i)**2
        If c in s:
            Return False
        Else:
            S.add(c)

    Return True
```

## LeetCode 第 290 题：

**题目描述**

给出一个模式(pattern)以及一个字符串，判断这个字符串是否符合模式

示例 1:
输入: pattern = "abba",
str = "dog cat cat dog"

输出: true
示例 2:
输入:pattern = "abba",
str = "dog cat cat fish"
输出: false
示例 3:
输入: pattern = "aaaa", str = "dog cat cat dog"
输出: false
示例 4:
输入: pattern = "abba", str = "dog dog dog dog"
输出: false

# 代码如下：

```
class Solution:
    def wordPattern(self,pattern, str):
        str = str.split()
        return list(map(pattern.index,pattern)) == list(map(str.index,str))
```

## LeetCode 第 205 题：

**题目描述**

给定两个字符串 s 和 t，判断它们是否是同构的。 如果 s 中的字符可以被替换得到 t ，那么这两个字符串是同构的。 所有出现的字符都必须用另一个字符替换，同时保留字符的顺序。两个字符不能映射到同一个字符上，但字符可以映射自己本身。

示例 1:
输入: s = "egg", t = "add"
输出: true
示例 2:
输入: s = "foo", t = "bar"
输出: false
示例 3:
输入: s = "paper", t = "title"
输出: true

```
class Solution:
    def isIsomorphic(self, s: str, t: str) -> bool:
        return list(map(s.index,s)) == list(map(t.index,t))
```

# LeetCode 第 451 题：

**题目描述**

给定一个字符串，请将字符串里的字符按照出现的频率降序排列。

示例 1:

输入:

"tree"

输出:

"eert"

示例 2:

输入:

"cccaaa"

输出:

"cccaaa"

示例 3:

输入:

"Aabb"

输出:

"bbAa"

```
class Solution:
    def frequencySort(self, s: str) -> str:
        From collections import Counter
        D=Counter(s)
        s=Sorted(D.items(), key=lambda item:item[1], reverse = True)
        res = ''
        for key, value in s:
            res += key * value
            return res
```