Leetcode(python版) 排序简单题

242有效的字母异位词

题目：给定两个字符串 *s* 和 *t* ，编写一个函数来判断 *t* 是否是 *s* 的一个字母异位词。

**示例 1:**

**输入:** *s* = "anagram", *t* = "nagaram"

**输出:** true

**示例 2:**

**输入:** *s* = "rat", *t* = "car"

**输出:** false

**说明:**  
你可以假设字符串只包含小写字母。

**进阶:**  
如果输入字符串包含 unicode 字符怎么办？你能否调整你的解法来应对这种情况？

思路：先比较长度应该就可以过一些数据了，然后因为是**逐个字母**检查的所以一旦发现不符合就可以退出来

代码：

1. **class** Solution(object):
2. **def** isAnagram(self, s, t):
3. """
4. :type s: str
5. :type t: str
6. :rtype: bool
7. """
8. **if** len(s)!=len(t):
9. **return** False
10. c=set(t)
11. **for** i **in** c:
12. **if** t.count(i)!=s.count(i):
13. **return** False
14. **return** True

349两个数组的交集

题目：给定两个数组，编写一个函数来计算它们的交集。

**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]

**说明:**

* 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。

思路：Set存入第一个数组的值   
遍历第二个数组，如果第二个的数在Set中出现，那么就是交集（与此同时，因为只能返回一个值，所以出现后还需要从Set中删除哦

代码：

1. **class** Solution(object):
2. **def** intersection(self, nums1, nums2):
3. """
4. :type nums1: List[int]
5. :type nums2: List[int]
6. :rtype: List[int]
7. """
8. set1 = set(nums1)
9. ans = []
10. **for** x **in** nums2:
11. **if** x **in** set1:
12. ans += x,
13. set1.remove(x)
14. **return** ans

350两个数组的交集II

题目：给定两个数组，编写一个函数来计算它们的交集。

**示例 1:**

**输入:** nums1 = [1,2,2,1], nums2 = [2,2]

**输出:** [2,2]

**示例 2:**

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

**输出:** [4,9]

**说明：**

* 输出结果中每个元素出现的次数，应与元素在两个数组中出现的次数一致。
* 我们可以不考虑输出结果的顺序

思路：

代码：

1. **class** Solution(object):
2. **def** intersect(self, nums1, nums2):
3. """
4. :type nums1: List[int]
5. :type nums2: List[int]
6. :rtype: List[int]
7. """
8. res=[]
9. dict={}
10. **for** num **in** nums1:
11. **if** num **not** **in** dict:
12. dict[num]=1
13. **else**:
14. dict[num]+=1
15. **for** num **in** nums2:
16. **if** num **in** dict **and** dict[num]>0:
17. dict[num]-=1
18. res.append(num)
19. **return** res