

## 第二周作业

---

### 1. 有效的字母异位词

```
1 class Solution:
2     def isAnagram(self, s: str, t: str) -> bool:
3         if len(s) != len(t):
4             return False
5         se = set(s)
6         if se == set(t):
7             for i in se:
8                 if s.count(i) != t.count(i):
9                     return False
10            return True
11        else:
12            return False
```

小结：set的使用方法。

### set:无序不重复的集合

集合 (set) 是一个无序的不重复元素序列。

可以使用大括号 { } 或者 set() 函数创建集合，注意：创建一个空集合必须用 set() 而不是 { }，因为 { } 是用来创建一个空字典。

set的内置方法：

<code>add()</code>	为集合添加元素
<code>clear()</code>	移除集合中的所有元素
<code>copy()</code>	拷贝一个集合
<code>difference()</code>	返回多个集合的差集
<code>difference_update()</code>	移除集合中的元素，该元素在指定的集合也存在。
<code>discard()</code>	删除集合中指定的元素
<code>intersection()</code>	返回集合的交集
<code>intersection_update()</code>	返回集合的交集。
<code>isdisjoint()</code>	判断两个集合是否包含相同的元素，如果没有返回 True，否则返回 False。
<code>issubset()</code>	判断指定集合是否为该方法参数集合的子集。
<code>issuperset()</code>	判断该方法的参数集合是否为指定集合的子集
<code>pop()</code>	随机移除元素
<code>remove()</code>	移除指定元素
<code>symmetric_difference()</code>	返回两个集合中不重复的元素集合。
<code>symmetric_difference_update()</code>	移除当前集合中在另外一个指定集合相同的元素，并将另外一个指定集合中不同的元素插入到当前集合中。
<code>union()</code>	返回两个集合的并集
<code>update()</code>	给集合添加元素

回答本题时：

- (1) 和面试官把题目过一下，确定本题的意思，如：什么是异位词？是否考虑大小写？
- (2) 本题可能的解法（最差也要会暴力），找到最优的解法。
- (3) coding.....
- (4) Test.....

## 2.字母异位词分组

给定一个字符串数组，将字母异位词组合在一起。字母异位词指字母相同，但排列不同的字符串。

```

输入：["eat", "tea", "tan", "ate", "nat", "bat"]
输出：
[
  ["ate","eat","tea"],
  ["nat","tan"],
  ["bat"]
]
```

解法一：（优秀的解法，看不明白。。。）

```

1 class Solution:
2     def groupAnagrams(self, strs: List[str]) -> List[List[str]]:
3         d = {}
4         for w in strs:
5             key = tuple(sorted(w))
6             d[key] = d.get(key, []) + [w]
7         return sorted(d.values())

```

解法二：

```

1 class Solution:
2     def groupAnagrams(self, strs: List[str]) -> List[List[str]]:
3         ans = collections.defaultdict(list)
4         for s in strs:
5             ans[tuple(sorted(s))].append(s)
6         return sorted(ans.values())
7 )

```

小结：defaultdict()的用法

(1) 使用 `list` 作为 `default_factory`，很轻松地将（键-值对组成的）序列转换为（键-列表组成的）字典

```

>>> s = [('yellow', 1), ('blue', 2), ('yellow', 3), ('blue', 4), ('red', 1)]
>>> d = defaultdict(list)
>>> for k, v in s:
...     d[k].append(v)
...
>>> sorted(d.items())
[('blue', [2, 4]), ('red', [1]), ('yellow', [1, 3])]

```

(2) 设置 `default_factory` 为 `int`，使 `defaultdict` 用于计数

```

>>> s = 'mississippi'
>>> d = defaultdict(int)
>>> for k in s:
...     d[k] += 1
...
>>> sorted(d.items())
[('i', 4), ('m', 1), ('p', 2), ('s', 4)]

```

(3) 设置 `default_factory` 为 `set` 使 `defaultdict` 用于构建 set 集合：

```

>>> s = [('red', 1), ('blue', 2), ('red', 3), ('blue', 4), ('red', 1), ('blue', 4)]
>>> d = defaultdict(set)
>>> for k, v in s:
...     d[k].add(v)
...
>>> sorted(d.items())
[('blue', {2, 4}), ('red', {1, 3})]

```