242-有效的字母异位词

题述

242. 有效的字母异位词

难度 简单 477

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。

注意：若 `s` 和 `t` 中每个字符出现的次数都相同，则称 `s` 和 `t` 互为字母异位词。

示例 1:

```
输入: s = "anagram", t = "nagaram"
输出: true
```

示例 2:

```
输入: s = "rat", t = "car"
输出: false
```

提示:

- `1 <= s.length, t.length <= 5 * 10^4`
- `s` 和 `t` 仅包含小写字母

思路

Way 1--排序然后比较 时间复杂度高

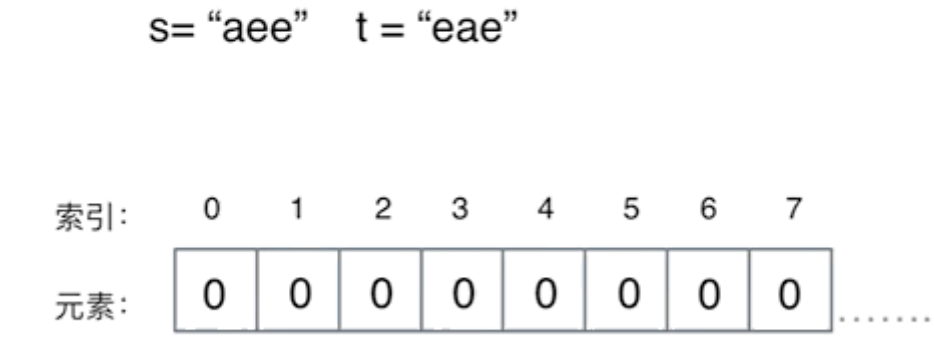
Way 2--哈希表法

数组其实就是一个简单哈希表，而且这道题目中字符串只有小写字符，那么就可以定义一个数组，来记录字符串s里字符出现的次数。

需要定义一个多大的数组呢，定一个数组叫做record，大小为26 就可以了，初始化为0，因为字符a到字符z的ASCII也是26个连续的数值。

为了方便举例，判断一下字符串s= "aee", t = "eae"。

操作动画如下：



定义一个数组叫做record用来上记录字符串s里字符出现的次数。

需要把字符映射到数组也就是哈希表的索引下标上，**因为字符a到字符z的ASCII是26个连续的数值，所以字符a映射为下标0，相应的字符z映射为下标25。**

再遍历 字符串s的时候，**只需要将 s[i] - ‘a’ 所在的元素做+1 操作即可，并不需要记住字符a的ASCII，只要求出一个相对数值就可以了。** 这样就将字符串s中字符出现的次数，统计出来了。

那看一下如何检查字符串t中是否出现了这些字符，同样在遍历字符串t的时候，对t中出现的字符映射哈希表索引上的数值再做-1的操作。

那么最后检查一下，**record数组如果有的元素不为零0，说明字符串s和t一定是谁多了字符或者谁少了字符，return false。**

最后如果record数组所有元素都为零0，说明字符串s和t是字母异位词，return true。

时间复杂度为$O(n)$，空间上因为定义是的一个常量大小的辅助数组，所以空间复杂度为$O(1)$。

题解

排序法

```
class Solution {
public:
    bool isAnagram(string s, string t)
    {
        //排序法
        if(s.length()!=t.length())
        {
            return false;
        }
        sort(s.begin(),s.end());
        sort(t.begin(),t.end());
        return s==t;
    }
};
```

哈希表法

```
class Solution {
public:
    bool isAnagram(string s, string t)
    {
        //哈希表法  数组是哈希表的一种
        int record[26]={};  //0-a  25-z

        //遍历两个字符串
        for(int i=0;i<s.size();i++)
        {
            //不用记住ASCII值 只需记录出现次数
            record[s[i]-'a']++;     //加加
        }

        for(int i=0;i<t.size();i++)
        {
            //不用记住ASCII值 只需记录出现次数
            record[t[i]-'a']--;     //减减
        }

        for(int i=0;i<26;i++)
        {
            if(record[i]!=0)
            {
                //record如果有的元素不为0  那么肯定不是字母异位词
                return false;
            }
        }
        return true;
    }
};
```

Python版本

```python
class Solution:
    def isAnagram(self, s: str, t: str) -> bool:
        #哈希表法
        record=[0]*26
        for i in range(len(s)):
            #不用记住字符的ASCII值 只需要记录差值
            record[ord(s[i])-ord("a")]+=1
        print(record)
        for i in range(len(t)):
            record[ord(t[i])-ord("a")]-=1
        for i in range(26):
            if record[i]!=0:
                return False
        return True
```

写Python版本是因为最近在温习Python

回顾

判断一个元素是否出现过的场景也应该第一时间想到哈希法