

0242. 有效的字母异位词

👤 ITCharge 🕒 大约 1 分钟

- 标签：哈希表、字符串、排序
- 难度：简单

题目链接

- [0242. 有效的字母异位词 - 力扣](#)

题目大意

描述：给定两个字符串 s 和 t 。

要求：判断 t 和 s 是否使用了相同的字符构成（字符出现的种类和数目都相同）。

说明：

- **字母异位词：**如果 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。
- $1 \leq s.length, t.length \leq 5 \times 10^4$ 。
- s 和 t 仅包含小写字母。

示例：

- 示例 1:

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

py

- 示例 2:

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

py

解题思路

思路 1：哈希表

1. 先判断字符串 s 和 t 的长度，不一样直接返回 `False`；
2. 分别遍历字符串 s 和 t 。先遍历字符串 s ，用哈希表存储字符串 s 中字符出现的频次；
3. 再遍历字符串 t ，哈希表中减去对应字符的频次，出现频次小于 0 则输出 `False`；
4. 如果没出现频次小于 0，则输出 `True`。

思路 1：代码

```
def isAnagram(self, s: str, t: str) -> bool:
    if len(s) != len(t):
        return False
    strDict = dict()
    for ch in s:
        if ch in strDict:
            strDict[ch] += 1
        else:
            strDict[ch] = 1
    for ch in t:
        if ch in strDict:
            strDict[ch] -= 1
            if strDict[ch] < 0:
                return False
        else:
            return False
    return True
```

py

思路 1：复杂度分析

- 时间复杂度： $O(n + m)$ ，其中 n 、 m 分别为字符串 s 、 t 的长度。
- 空间复杂度： $O(|S|)$ ，其中 S 为字符集大小，此处 $S == 26$ 。

