

0205. 同构字符串

👤 ITCharge 🕒 大约 2 分钟

- 标签：哈希表、字符串
- 难度：简单

题目链接

- [0205. 同构字符串 - 力扣](#)

题目大意

描述：给定两个字符串 s 和 t 。

要求：判断字符串 s 和 t 是否是同构字符串。

说明：

- **同构字符串：**如果 s 中的字符可以 一种映射关系替换得到 t 相同位置上的字符，那么两个字符串是同构的。
- 每个字符都应当映射到另一个字符，且不改变字符顺序。不同字符不能映射到统一字符上，相同字符只能映射到同一个字符上，字符可以映射到自己本身。
- $1 \leq s.length \leq 5 \times 10^4$ 。
- $t.length == s.length$ 。
- s 和 t 由任意有效的 ASCII 字符组成。

示例：

- 示例 1：

输入： $s = \text{"egg"}, t = \text{"add"}$

输出：True

py

- 示例 2：

输入： $s = \text{"foo"}, t = \text{"bar"}$

py

输出: `False`

解题思路

思路 1: 哈希表

根据题目意思, 字符串 s 和 t 每个位置上的字符是一一对应的。 s 的每个字符都与 t 对应位置上的字符对应。可以考虑用哈希表来存储 $s[i] : t[i]$ 的对应关系。但是这样不能只能保证对应位置上的字符是对应的, 但不能保证是唯一对应的。所以还需要另一个哈希表来存储 $t[i] : s[i]$ 的对应关系来判断是否是唯一对应的。

思路 1: 代码

```
class Solution:
    def isIsomorphic(self, s: str, t: str) -> bool:
        s_dict = dict()
        t_dict = dict()
        for i in range(len(s)):
            if s[i] in s_dict and s_dict[s[i]] != t[i]:
                return False
            if t[i] in t_dict and t_dict[t[i]] != s[i]:
                return False
            s_dict[s[i]] = t[i]
            t_dict[t[i]] = s[i]
        return True
```

py

思路 1: 复杂度分析

- 时间复杂度: $O(n)$, 其中 n 为字符串长度。
- 空间复杂度: $O(|S|)$, 其中 S 是字符串字符集。