

0014. 最长公共前缀

👤 [ITCharge](#) ⌚ 大约 1 分钟

- 标签：字典树、字符串
- 难度：简单

题目链接

- [0014. 最长公共前缀 - 力扣](#)

题目大意

描述： 给定一个字符串数组 `strs` 。

要求： 返回字符串数组中的最长公共前缀。如果不存在公共前缀，返回空字符串 `""` 。

说明：

- $1 \leq \text{strs.length} \leq 200$ 。
- $0 \leq \text{strs}[i].\text{length} \leq 200$ 。
- `strs[i]` 仅由小写英文字母组成。

示例：

- 示例 1:

```
输入: strs = ["flower","flow","flight"]  
输出: "fl"
```

py

- 示例 2:

```
输入: strs = ["dog","racecar","car"]  
输出: ""  
解释: 输入不存在公共前缀。
```

py

解题思路

思路 1：纵向遍历

- 依次遍历所有字符串的每一列，比较相同位置上的字符是否相同。
 - 如果相同，则继续对下一列进行比较。
 - 如果不相同，则当前列字母不再属于公共前缀，直接返回当前列之前的部分。
- 如果遍历结束，说明字符串数组中的所有字符串都相等，则可将字符串数组中的第一个字符串作为公共前缀进行返回。

思路 1：代码

```
class Solution:
    def longestCommonPrefix(self, strs: List[str]) -> str:
        if not strs:
            return ""

        length = len(strs[0])
        count = len(strs)
        for i in range(length):
            c = strs[0][i]
            for j in range(1, count):
                if len(strs[j]) == i or strs[j][i] != c:
                    return strs[0][:i]
        return strs[0]
```

py

思路 1：复杂度分析

- 时间复杂度：** $O(m \times n)$ ，其中 m 是字符串数组中的字符串的平均长度， n 是字符串的数量。
- 空间复杂度：** $O(1)$ 。