Leetcode: 14
Longest Common Prefix

14. Longest Common Prefix

Easy | Topics | Companies

Write a function to find the longest common prefix string amongst an array of strings.

If there is no common prefix, return an empty string `""`.

Example 1:

```
Input: strs = ["flower","flow","flight"]
Output: "fl"
```

Example 2:

```
Input: strs = ["dog","racecar","car"]
Output: ""
Explanation: There is no common prefix among the input strings.
```

Constraints:

- `1 <= strs.length <= 200`
- `0 <= strs[i].length <= 200`
- `strs[i]` consists of only lowercase English letters if it is non-empty.

Constraints:

- `1 <= nums.length <= 3 * 10^4`
- `-100 <= nums[i] <= 100`
- `nums` is sorted in **non-decreasing** order.

Explanation

longest common prefix

flower, flow, flight

flower, flow, flight

shortest = flow
shortest = fl

Code

```python
class Solution:
    def longestCommonPrefix(self, strs: List[str]) -> str:
        shortest = min(strs, key=len)
        shortestLength = len(shortest)

        for i in strs:
            for j in range(shortestLength):
                if i[j] == shortest[j]:
                    continue
                else:
                    shortest = i[:j]
                    shortestLength = len(shortest)
                    break
        return shortest
```

Code

Python3 Auto

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

        # Compare character by character across all strings
        for i in range(len(strs[0])):
            char = strs[0][i]
            for j in range(1, len(strs)):
                # Check if we've reached end of current string or found mismatch
                if i >= len(strs[j]) or strs[j][i] != char:
                    return strs[0][:i]

        return strs[0]


```

VMES Grinds

Time and Space Complexity Analysis

```python
class Solution:
    def longestCommonPrefix(self, strs: List[str]) -> str:
        shortest = min(strs, key=len)
        shortestLength = len(shortest)

        for i in strs:
            for j in range(shortestLength):
                if i[j] == shortest[j]:
                    continue
                else:
                    shortest = i[:j]
                    shortestLength = len(shortest)
                    break
        return shortest
```

Space - O(m)

Time - O(nxm)