[Click on the logo to find **Problem Statement**]

Intuition

This code aims to find the longest common prefix among a list of strings. It begins by sorting the input list of strings, which facilitates the comparison of the common prefix between the first and last strings after sorting. By examining the characters at each position in the first and last strings, the algorithm iteratively builds the longest common prefix. If a mismatch is encountered, the current prefix is returned. Otherwise, the matching character is appended to the result. This process continues until the end of the shorter string is reached or a mismatch occurs. The sorted order ensures that the first and last strings represent the lexicographically smallest and largest strings in the original list, aiding in the efficient identification of the common prefix.

Approach

1. Edge Case Check:
    - If the input list strs is empty, return an empty string as there is no common prefix.
2. Initialize Prefix:
    - Initialize a variable prefix with the first string in the list strs[0].This string will serve as the initial assumed common prefix.
3. Iterate through Strings:
    - Iterate through the remaining strings in the list (from strs[1] onwards).
4. Find Common Prefix:
    - For each string in the list, iterate through the characters of the assumed common prefix and the current string.
    - Compare characters at the same position.
    - If a mismatch is found or if the index exceeds the length of the current string, update the prefix variable to be the substring up to the current index and break out of the loop. This is because any characters beyond this index cannot be part of the common prefix.
5. Return Result:

- After iterating through all strings, the prefix variable will contain the longest common prefix. Return prefix as the result.

Complexity

- Time complexity: O(m * n * log n)
- Space complexity: O(1)

Code

```
class Solution:
    def longestCommonPrefix(self, strs: List[str]) -> str:
        ans = ""
        strs = sorted(strs)
        start = strs[0]
        end = strs[-1]
        for i in range(min(len(start), len(end))):
            if start[i] != end[i]:
                return ans
            ans += start[i]
        return ans
```

If you want to see more solutions to coding problems, you can visit: