

Intuition

If the first characters of the strings all match compare the first 2 characters

...and so on all the way to the first M characters. Return the last matching prefix in the sequence.

Approach

Slice the first element of the list to its first k characters. Multiply it to form a list so it can be compared to the input list with each element sliced to the first k characters. If these list match then increment k

...and so on to M if so.

Complexity

- Time complexity:

$$O(M * N)$$

Length of shortest string and size of list

- Space complexity:

$$O(M * N)$$

Code

```
class Solution:
```

```

def longestCommonPrefix(self, strs: List[str]) -> str:
    common = ""
    c = 1
    while c <= 200:
        if ([strs[0][:c]] * len(strs)) == list(map(lambda x:
x[:c], strs)):
            common = strs[0][:c]
            c += 1
        else:
            return common

    return common

```