Longest Common prefix. Leetcode

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 "".

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

Program:

```
class Solution{
    public String longestCommonPrefix(String [] strs){
        String result = "";
        int currIndex =0;
        if (strs.length ==1)
            return str[0];
        while (true) {
            for (int i=0; i< strs.length -1 ;i++)
        {
         if(currIndex >= str[i].length() ||
currIndex >= strs[i+1]. length() || strs[i]. charAt(currIndex)
!= strs[i+1]. charAt (currIndex)) {
                    return result;
                    }}
        result = result + strs[0] .charAt(currIndex);
        currIndex +=1;}}}.
```

```
Class Solution {
    public String longestCommonPrefix(String [] strs) {
      if (strs == null || strs.length ==0){
            return "";
      }
    if (strs.length ==1){
          return strs[0];
    }
    String result = "";
    int currIndex =0;
    while (true){
        for(int i=0; i< strs.length -1; i++){
          if(currIndex >= strs[i].length||
            currIndex >= strs[i+1].length() ||
            strs[i].charAt(currIndex) !=
              strs[i+1].charAt(currIndex ){
                return result;}}
          result += str[0].charAt(currIndex);
          currIndex ++; }}}.
```

str = {"flower", "flow", "flight"}

Initialization

check "" "".

strs == null || strs.length == 0.

false false. → false.

check

strs.length == 1

3 == 1 ✗ → false.

result = ""

currIndex = 0.

While loop Execution

Iteration (1) currIndex = 0.

Enter while loop

start the for loop.

for p = 0. p < strs.length - 1

p < 3 - 1 p < 2. 0 < 2 ✓

p && (currIndex >= strs[p].length())

0 >= strs[0].length() 0 >= 6 ✗

currIndex >= strs[p+1].length()

0 >= strs[1].length() 0 >= 4 ✗.

strs[p].charAt(currIndex) != strs[p+1].charAt(currIndex))

strs[0].charAt(0) = f

str[1].charAt(0) = f f != f ✗.

(match → continue)

p = 1 p < strs.length - 1

p < 3 - 1 1 < 2 ✓

if (currIndex >= strs[p].length())

0 >= 4 ✗

currIndex >= strs[p+1].length())

0 >= 6 ✗.

strs[p].charAt(currIndex) != strs[p+1].charAt(currIndex))

strs[1].charAt(0) = 'f' X.
strs[2].charAt(0) = 'f'
Match -> Exit for loop.
↳ Because p=0 p < strs.length - 1
p < 2. 3 < 2 X.

result += strs[0].charAt(currIndex);
result += strs[0].charAt(0);
result = result + 0
result = "" + f
result = f
currIndex++; currIndex = 1

Output
f.

Question 2.

Enter while loop.
Start the for loop
for p=0 p < str.length - 1
p < 3-1 p < 2 0 < 2 ✓

If
(currIndex >= strs[p].length()
1 >= 6 X.
(currIndex >= strs[p+1].length()
1 >= strs[1].length() 1 >= 4 X.
strs[p].charAt(currIndex) !=
strs[p+1].charAt(currIndex])
strs[0].charAt(1) !=
strs[p+1].charAt(currIndex)

fl != fl x.
match continue.

p=1 p < strs.length-1 p < 3-1 1 < 2 ✓.

if (currIndex >= strs[p].length())
0 >= 4 X
currIndex >= strs[p+1].length()
0 >= 6 X.
strs[p].charAt(currIndex) != strs[p+1].charAt(currIndex))
strs[1].charAt(1) = 'l'
strs[2].charAt(1) = 'l' → match → Exit for loop.
↳ Because p=3 p < strs.length-1 p < 2 3 ≮ 2 X.

result += strs[0].charAt(currIndex);
result += strs[0].charAt(1);
result = f + l
result = fl
currIndex++;
currIndex = 1

o/p fl.

Iteration 3

Enter while loop start the for loop.
for p=0 p < str.length-1
p < 3-1 p < 2 0 < 2 ✓.
if (currIndex >= strs[p].length())
1 >= 6 X
(currIndex >= strs[p+1].length()
1 >= strs[].length 1 >= 4 X.
strs[p].charAt(currIndex) !=
strs[p+1].charAt(currIndex))
strs[0].charAt(2) !=
strs[p].charAt(2) 0 != 0 X

mateh continue.

p=1

p < strs.length - 1

p < 3 - 1

p < 2 → 1

For (currIndex >= strs[p].length())

0 >= 4 ×

currIndex >= strs[p+1].length()

0 >= 6 ×

strs[p].charAt(currIndex) != strs[p+1].charAt(currIndex)

strs[1].charAt(2) = 0

strs[2].charAt(2) = p.

↳ Not match exit the loop.

result = ""