

14. Longest Common Prefix

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:

- $1 \leq \text{strs.length} \leq 200$
- $0 \leq \text{strs[i].length} \leq 200$
- strs[i] consists of only lowercase English letters.

Longest Common Prefix

["VISHAL"] ["VISHU"]

["VISH"]

{ we have to return }
this string

Constraints:

Strings max: 200

every string length max: 200

String consists only lower case letters

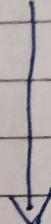
Approach

: On the first string traverse horizontally along with the remaining strings but vertically

$\xrightarrow{[V I S H] A L]$

$[V I S H U]$

$[V I S H]$



Traverse : 1

$$[V] = \begin{bmatrix} V \\ V \end{bmatrix}$$

True

Traverse : 4

True

longest prefix = V

$$H = \begin{bmatrix} H \\ H \end{bmatrix}$$

Traverse : 2

$$IP = VISH$$

$$[I] = \begin{bmatrix} I \\ I \end{bmatrix}$$

True

Traverse : 5

longest prefix = VI

(One string
is over
false)

Traverse : 3

$$[S] = \begin{bmatrix} S \\ S \end{bmatrix}$$

True

longest pref = VIS

Code: (Create a string of longest prefix)

- 1) make a variable and store total number of strings.

int nofstrings = str.length;

- 2) make a variable who holds 0th string length.

int size = str[0].length();

- 3) Run first for loop horizontally to str[8]. and also create a variable to holds its char.

[For (int i=0; i<size; i++)
 char value = str[0].charAt(i);]

- 4) Now, Run a for loop inside this upper ↑ loop but vertically and compare this {value} to every string chars.

: we start it from 1 because 0 string is taken for max.

for (int j=1; j<nofstrings; j++) {

 if (value != str[j].charAt(i)) {

 return i;
 }

CLASSTIME _____
Date / /

⑤ Outside the for loop add the char into IP because it presents in all the strings.

for (loop1 →) {

Value =

for (loop2 ↓) {

[IP += ~~value~~;]

Now, full code:

For (loop for hori → traversing) {

Value i = str[i].charAt()

for ((loop for vertical ↓ traversing)) {

if (value != str[i].charAt(i)) {

return IP;

}

IP += value;

}

return IP;

Special case:

⇒ if a string is ends or shorter than the horizontal string

examples:

① [ä b] [a]

② ["ä bc"] ["äb"] ["ä"]

⇒ for solving this we also check for this condition

if ($\text{str}[i].length() - 1 < i$) {

 return 16;

}

mean, if current string length is smaller than the current index on which we are checking on the return the 16; ~~because~~