

Program Name- Longest Common Prefix | Set 1 (Word by Word Matching)

Project Category- Strings

Programming Paradigm Used- Associative Property of Longest Common Prefix Function

Algorithm / Explanation-

We start with an example. Suppose there are two strings- “geeksforgeeks” and “geeks”. What is the longest common prefix in both of them?

It is “geeks”.

Now let us introduce another word “geek”. So now what is the longest common prefix in these three words ?

It is “geek”

We can see that the longest common prefix holds the associative property, i.e-

$LCP(\text{string1}, \text{string2}, \text{string3}) = LCP(LCP(\text{string1}, \text{string2}), \text{string3})$

Like here

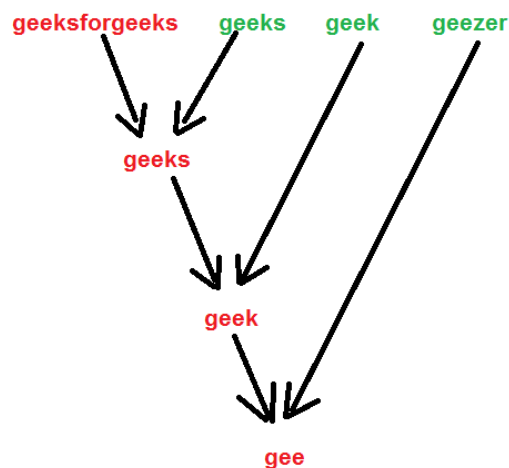
$LCP(\text{“geeksforgeeks”}, \text{“geeks”}, \text{“geek”}) = LCP(LCP(\text{“geeksforgeeks”}, \text{“geeks”}), \text{“geek”}) = LCP(\text{“geeks”}, \text{“geek”}) = \text{“geek”}$

So we will make use of the above associative property to find the LCP of the given strings.

We one by one calculate the LCP of each of the given string with the LCP so far. The final result will be our longest common prefix of all the strings.

Note that it is possible that the given strings have no common prefix. This happens when the first character of all the strings are not same.

We show the algorithm with the input strings- “geeksforgeeks”, “geeks”, “geek”, “geezer” by the below figure-



--> Words in **Red** are the longest prefix string at each iteration.

--> Our final result is "**gee**" which is the longest prefix string among "**geeksforgeeks**", "**geeks**", "**geek**" and "**geezer**"

Time Complexity-

Since we are iterating through all the strings and for each string we are iterating through each character, so we can say that the time complexity is $O(NM)$ where,

N = Number of strings

M = Average length of each string

Auxiliary Space-

To store the longest prefix string we are allocating space which is $O(N)$ where,

N = length of the largest string among all the strings

How to improve this ?-

In the next few sets we will discuss about how to improve the algorithm to find the longest common prefix string.