Program Name- Longest Common Prefix | Set 5 (Trie)

Project Category- Strings

Data Structure Used- Trie

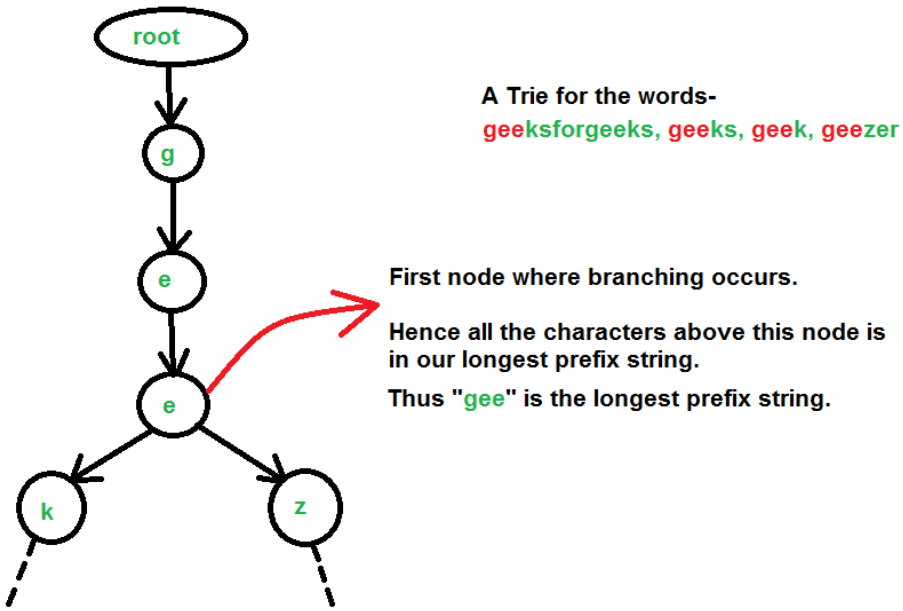
Algorithm/Explanation-

We will insert all the words one by one in the trie. After inserting we perform a walk on the trie.

In this walk, we will go deeper until we find a node having more than 1 children(branching occurs) or 0 children (one of the string gets exhausted).

This is because the characters (nodes in trie) which are present in the longest common prefix must be the single child of its parent, i.e- there should not be a branching in any of these nodes.

The algorithm will be clear using the below illustration. We consider our strings as - “geeksforgeeks”, “geeks”, “geek”, “geezer”



Time Complexity-

Inserting all the words in the trie takes O(MN) time and performing a walk on the trie takes O(K) time, where-

N = number of strings,
M = avg. length of each string,
K = length of the longest string in the input array

Since K <= NM (equality when there is only one string in our input array)

So, the overall time complexity is O(MN)

Auxiliary Space-

To store all the strings we need to allocate O(26*M*N) ~ O(MN) space for the trie where-

N = number of strings,
M = avg. length of each string,