Topics Tags : Trie, String,Sorting

Longest Common Prefix : You are given an array of N strings. You need to find the longest common prefix among all strings present in the array (in case there is no common prefix return -1).

Example 1:

Input:
N = 4
ar[] = {geeksforgeeks, geeks, geek, geezer}

Output:
gee

Explanation:
Longest common prefix in all the given string is gee.

Expected time complexity: O(NlogN)

Expected space complexity: O(string length)

Constraints:

1 <= N <= 10^3

1 <= String Length <= 100

```
class TrieNode{

    TrieNode[] children;
    char data;
    int count;

    public TrieNode(char data){
        this.children = new TrieNode[26];
        this.data = data;

        for(int i = 0; i < 26; ++i)
            this.children[i] = null;

        this.count = 0;
    }

    public TrieNode subNode(int n){

        for(TrieNode child : this.children){

            if(child == null)
                continue;

            if(child.count == n)
                return child;
        }

        return null;
    }
}

class Solution {

    private TrieNode root = new TrieNode('_');

    private void insert(String key){
         TrieNode node = root;

        for(char ch : key.toCharArray()){
         int idx = ch - 'a';
```

```
            if(node.children[idx] == null)
                node.children[idx] = new TrieNode(ch);

            node = node.children[idx];
            node.count += 1;
        }

        return;
    }

    private String findCommonPrefix(int n){

        TrieNode node = root;
        StringBuilder sb = new StringBuilder();

        while(true){
            TrieNode child = node.subNode(n);

            if(child == null)
                break;

            sb.append(child.data);
            node = child;
        }
        return sb.toString();
    }

    public String lcp(String arr[], int n) {
        // Write your code here

        for(String s : arr){
            insert(s);
        }

        String ans = findCommonPrefix(n);
        return ans.length() == 0 ? "-1" : ans;
    }
}
```