

14. Longest Common Prefix

Easy

Topics

Companies

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 <= strs.length <= 200`
- `0 <= strs[i].length <= 200`
- `strs[i]` consists of only lowercase English letters if it is non-empty.

Java

```
class Solution {
    public String longestCommonPrefix(String[] strs) {
        if (strs == null || strs.length == 0) return "";

        // Take the first string as the initial prefix
        String prefix = strs[0];

        // Compare with every other string
        for (int i = 1; i < strs.length; i++) {
            // Reduce prefix until it matches the beginning of strs[i]
            while (strs[i].indexOf(prefix) != 0) {
                prefix = prefix.substring(0, prefix.length() - 1);
            }
        }
    }
}
```

```

        // If prefix becomes empty, no common prefix
        if (prefix.isEmpty()) {
            return "";
        }
    }
}
return prefix;
}
}

```

JavaScript

```

/**
 * @param {string[]} strs
 * @return {string}
 */
var longestCommonPrefix = function(strs) {
    if (strs.length === 0) return "";

    // Start with the first string as prefix
    let prefix = strs[0];

    // Compare prefix with each string
    for (let i = 1; i < strs.length; i++) {
        while (strs[i].indexOf(prefix) !== 0) {
            // Reduce prefix length until match
            prefix = prefix.substring(0, prefix.length - 1);
            if (prefix === "") return "";
        }
    }

    return prefix;
};

```

Python

```

class Solution(object):
    def longestCommonPrefix(self, strs):
        if not strs:
            return ""

        # Start with the first string as prefix
        prefix = strs[0]

        for s in strs[1:]:
            while s[:len(prefix)] != prefix:

```

```
    prefix = prefix[:-1] # reduce prefix
    if not prefix:
        return ""
    return prefix
```