Longest Common Prefix

Write a function to find the longest common prefix string amongst an array of strings.

Solution 1

```java
public String longestCommonPrefix(String[] strs) {
    if(strs == null || strs.length == 0)    return "";
    String pre = strs[0];
    int i = 1;
    while(i < strs.length){
        while(strs[i].indexOf(pre) != 0)
            pre = pre.substring(0,pre.length()-1);
        i++;
    }
    return pre;
}
```

written by desmile original link here

Solution 2

Sort the array first, and then you can simply compare the first and last elements in the sorted array.

```java
public String longestCommonPrefix(String[] strs) {
    StringBuilder result = new StringBuilder();

    if (strs!= null && strs.length > 0){

        Arrays.sort(strs);

        char [] a = strs[0].toCharArray();
        char [] b = strs[strs.length-1].toCharArray();

        for (int i = 0; i < a.length; i ++){
            if (b.length > i && b[i] == a[i]){
                result.append(b[i]);
            }
            else {
                return result.toString();
            }
        }
    }
    return result.toString();
}
```

written by cassandra9 original link here

Solution 3

```cpp
class Solution {
public:
    string longestCommonPrefix(vector<string>& strs) {
        string prefix = "";
        for(int idx=0; strs.size()>0; prefix+=strs[0][idx], idx++)
            for(int i=0; i<strs.size(); i++)
                if(idx >= strs[i].size() ||(i > 0 && strs[i][idx] != strs[i-1][idx]))
                    return prefix;
        return prefix;
    }
};
```

written by huwenboshi original link here