

## 1.Easy\04.Longest\_common\_prefix.cpp

```
1  /*
2  Question:
3  Write a function to find the longest common prefix string amongst an array of strings.
4  If there is no common prefix, return an empty string "".
5
6  Approach:
7  1. Sort the array of strings lexicographically.
8  2. Take the first and last string from the sorted array.
9  3. Compare each character of the first and last string until they don't match or the end of
   either string is reached.
10 4. Return the common prefix.
11
12 Time Complexity:  $O(N*M*\log(N))$ , where N is the number of strings and M is the maximum length
   of the strings.
13 - Sorting the array of strings takes  $O(N*\log(N))$  time.
14 - Comparing the first and last string takes  $O(M)$  time.
15
16 Space Complexity:  $O(1)$ 
17 - We use constant space.
18
19 Code:
20 */
21
22 using namespace std;
23
24 string longestCommonPrefix(vector<string>& strs) {
25     if (strs.empty())
26         return "";
27
28     sort(strs.begin(), strs.end());
29
30     int first = 0, last = strs.size() - 1;
31     int i = 0, j = 0;
32     int len = 0;
33
34     while (i < strs[first].size() && j < strs[last].size() && strs[first][len] == strs[last]
35 [len]) {
36         i++;
37         j++;
38         len++;
39     }
40
41     return strs[first].substr(0, len);
42 }
```