**14.Longest Common Prefix**

**Approach 1: HashMap**

1. For every substring of a string, add it to hashMap
2. If any substring has a map frequency greater than max-common, update max-common and prefix to the substring
3. Check if final max-common is equivalent to number of strings in the array, that means substring has appeared in every word
4. IF so, return the substring
5. Otherwise return “”

Time -> O(m\*n)

Space -> O(longest word)

**Code:**

hashMap = defaultdict(lambda:0)

prefix = ""

max\_common = 0

for s in strs:

word = ""

for letter in s:

word += letter

hashMap[word] += 1

if hashMap[word] >= max\_common:

max\_common = hashMap[word]

prefix = word

if max\_common >= len(strs):

return prefix

return ""