Given an array of strings strs, group **the anagrams** together. You can return the answer in **any order**.

An **Anagram** is a word or phrase formed by rearranging the letters of a different word or phrase, typically using all the original letters exactly once.

**Example 1:**

**Input:** strs = ["eat","tea","tan","ate","nat","bat"]

**Output:** [["bat"],["nat","tan"],["ate","eat","tea"]]

**Example 2:**

**Input:** strs = [""]

**Output:** [[""]]

**Example 3:**

**Input:** strs = ["a"]

**Output:** [["a"]]

class Solution {

public List<List<String>> groupAnagrams(String[] strs) {

List<List<String>>ans=new ArrayList();

Map<String,List<String>> mp=new HashMap();

for(String s:strs)

{

char[] ch=s.toCharArray();

Arrays.sort(ch);

String tmp=new String(ch);

if(mp.containsKey(tmp))

mp.get(tmp).add(s);

else

{

List<String> a=new ArrayList();

a.add(s);

mp.put(tmp, a);

}

}

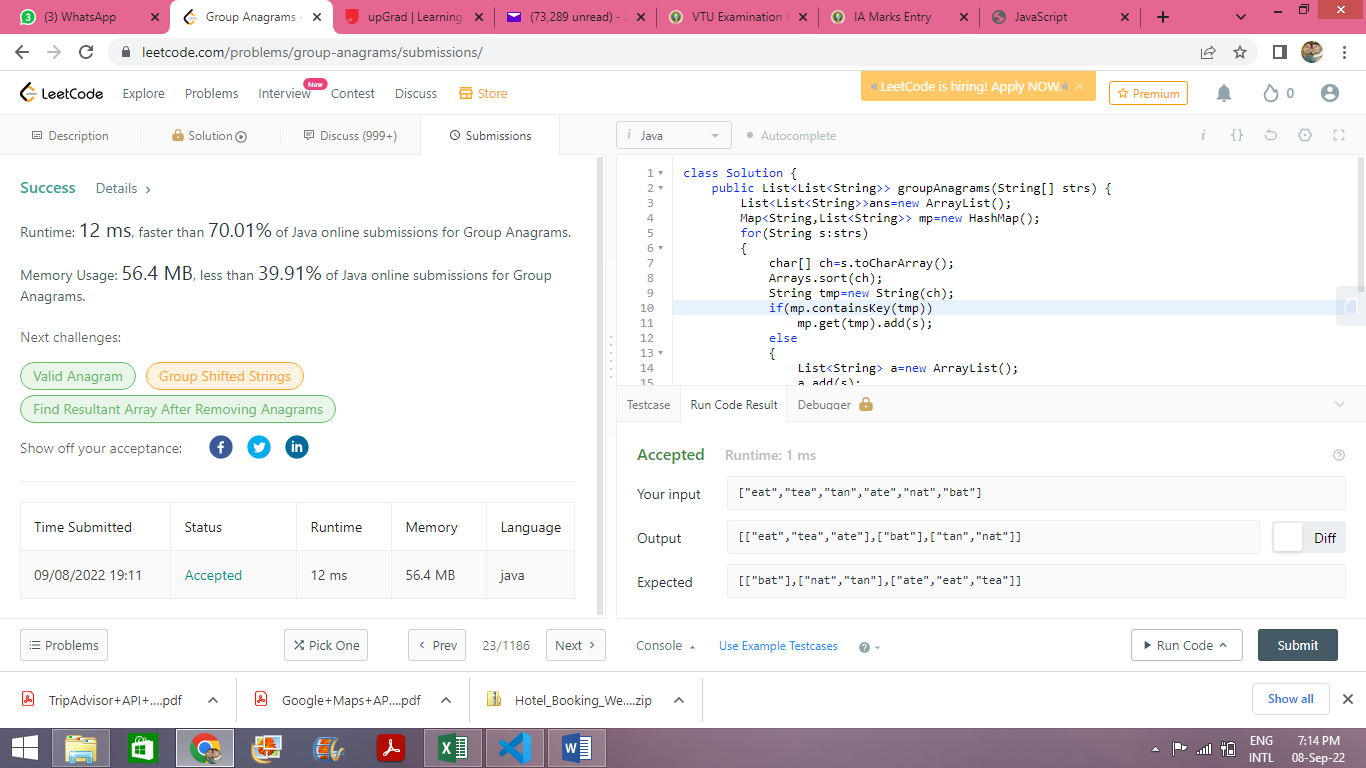
for(List<String> v:mp.values())

ans.add(v);

return ans;

}

}

z

class Solution {

public List<List<String>> groupAnagrams(String[] strs) {

List<List<String>>ans=new ArrayList();

Map<String,List<String>> mp=new HashMap();

for(String s:strs)

{

char[] ch=s.toCharArray();

Arrays.sort(ch);

String tmp=new String(ch);

if(mp.containsKey(tmp))

mp.get(tmp).add(s);

else

{

List<String> a=new ArrayList();

a.add(s);

mp.put(tmp, a);

}

}

for(List<String> v:mp.values())

ans.add(v);

for(List<String> l:ans)

System.out.println(l);

return ans;

}

}

