

Group Anagrams

Try to solve the Group Anagrams problem.

We'll cover the following ^

- Statement
- Examples
- Understand the problem
- Figure it out!
- Try it yourself

Statement

Given a list of words or phrases, group the words that are anagrams of each other. An **anagram** is a word or phrase formed from another word by rearranging its letters.

Constraints:

Let `strs` be the list of strings given as input to find the anagrams.

- $1 \leq \text{strs.length} \leq 10^4$
- $0 \leq \text{strs}[i].\text{length} \leq 100$
- `strs[i]` consists of lowercase English letters.

Note: The order in which the output is displayed doesn't matter.

Examples

Sample example 1

Input

strs	"duel"	"dule"	"speed"	"spede"	"deul"	"cars"
------	--------	--------	---------	---------	--------	--------



Output

output	[["duel", "dule", "deul"], ["speed", "spede"], ["cars"]]
--------	--

1 of 2

Understand the problem

Let's take a moment to make sure you've correctly understood the problem. The quiz below helps you check if you're solving the correct problem:

Group Anagrams

1

What is the output if the following list is given as input?

['bat', 'tab', 'tan', 'at']

A) [['at'], ['bat'], ['tab'], ['tan']]

B) [['at', 'bat']], [['tab', tan']]

C) [['at', 'bat', 'tab', tan']]

D) [['tab', 'bat']], ['tan'], ['at']]

Submit Answer

<


Question 1 of 3
0 attempted

>

Reset Quiz ↻

Figure it out!

We have a game for you to play. Rearrange the logical building blocks to develop a clearer understanding of how to solve this problem.

 Drag and drop the cards to rearrange them in the correct sequence.

Initialize a hash map to store key-value pairs for the strings' frequency and its anagrams, respectively. The key will be a character list of length 26, initialized to all 0s, and the value will be an array of anagrams.

Start traversing the list of strings. Within each string in the list, traverse each character of the string. Reset the character list to all 0s

?

Tt

🔄

before beginning this second traversal.

Calculate the index of each letter of the string through the ASCII value of the character. In the character list, increment the value at this index by 1.

After a string has been traversed, check whether the current character list is present as a key in the hash map.

If the current character list is present as a key in the hash map, add the string to the list of anagrams corresponding to this key.

Otherwise, add a new key-value pair to the hash map with the current character array as the key and the traversed string as the value in an array.

Repeat this process until all the strings have been traversed.

Reset

Show Solution

Submit



Implement your solution in the following coding playground:

Java

usercode > GroupAnagrams.java

```
1  import java.util.*;
2
3  class GroupAnagrams {
4
5      public static List<List<String>> groupAnagrams(String[] str){
6
7          // Write your code here
8          return new ArrayList<List<String>>();
9      }
10 }
```

Powered by AI

Submit

Test Cases

Results

Case 1

Case 2

Case 3

Input #1

["eat","tea","tan","ate","nat","bat"]

Group Anagrams

← Back

Solution: Design Tic-T...

Next →

Solution: Group Anagr...

☒ Mark as Completed