

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1	Solution 2	Solution 3
<pre>1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved. 2 3 import java.util.*; 4 import java.util.stream.*; 5 6 class Program { 7 // O(w * n * log(n)) time O(wn) space - where w is the number of words and n is the longest word 8 // the longest word 9 public static List<List<String>> groupAnagrams(List<String> words) { 10 Map<String, List<String>> anagrams = new HashMap<String, List<String>>(); 11 12 for (String word : words) { 13 char[] charArray = word.toCharArray(); 14 Arrays.sort(charArray); 15 String sortedWord = new String(charArray); 16 17 if (anagrams.containsKey(sortedWord)) { 18 anagrams.get(sortedWord).add(word); 19 } else { 20 anagrams.put(sortedWord, new ArrayList<String>(Arrays.asList(word))); 21 } 22 } 23 24 List<List<String>> output = new ArrayList<List<String>>(); 25 for (Map.Entry<String, List<String>> entry : anagrams.entrySet()) { 26 output.add(entry.getValue()); 27 } 28 return output; 29 } 30 } 31</pre>		<pre>1 import java.util.*; 2 3 class Program { 4 public static List<List<String>> groupAnagrams(List<String> words) { 5 // Write your code here. 6 return null; 7 } 8 } 9</pre>

Run or submit code when you're ready.