Our Solution(s)　　　　　　　　　　　　　　Run Code

Your Solutions　　　　　　　　　　　　　　Run Code

Solution 1　　Solution 2

Solution 1　　Solution 2　　Solution 3

```csharp
1   // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3   using System;
4   using System.Linq;
5   using System.Collections.Generic;
6
7   public class Program {
8     // O(w * n * log(n) + n * w * log(w)) time | O(wn) space - whe
9     // n is the length of the longest word
10    public static List<List<string> > groupAnagrams(List<string> w
11      if (words.Count == 0) return new List<List<string> >();
12
13      List<string> sortedWords = new List<string>();
14      foreach (string word in words) {
15        char[] charArray = word.ToCharArray();
16        Array.Sort(charArray);
17        string sortedWord = new String(charArray);
18        sortedWords.Add(sortedWord);
19      }
20
21      List<int> indices = Enumerable.Range(0, words.Count).ToList(
22      indices.Sort((a, b) => sortedWords[a].CompareTo(sortedWords[
23
24      List<List<string> > result = new List<List<string> >();
25      List<string> currentAnagramGroup = new List<string>();
26      string currentAnagram = sortedWords[indices[0]];
27      foreach (int index in indices) {
28        string word = words[index];
29        string sortedWord = sortedWords[index];
30
31        if (sortedWord.Equals(currentAnagram)) {
32          currentAnagramGroup.Add(word);
33          continue;
```

```csharp
1   using System.Collections.Generic;
2
3   class Program {
4     public static List<List<string> > groupAnagrams(List<string> w
5       // Write your code here.
6       return null;
7     }
8   }
9
```

Our Tests

Custom Output　　　　　　　　　　　　　　Submit Code

Run or submit code when you're ready.