

PromptScratchpadOur Solution(s)Video ExplanationRun Code

Solution 1Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class Program {
4     // O(w * n * log(n)) time | O(wn) space - where w is the number of words and
5     // n is the length of the longest word
6     func groupAnagrams(_ words: [String]) -> [[String]] {
7         var anagrams = [String: [String]]()
8
9         for word in words {
10             let sortedWord = String(word.sorted())
11             if var arr = anagrams[sortedWord] {
12                 arr.append(word)
13                 anagrams[sortedWord] = arr
14                 continue
15             }
16             anagrams[sortedWord] = [word]
17         }
18
19         var result = [[String]]()
20         for group in anagrams {
21             result.append(group.value)
22         }
23         return result
24     }
25 }
26
```

Your SolutionsRun Code

Solution 1Solution 2Solution 3

```
1 class Program {
2     func groupAnagrams(_ words: [String]) -> [[String]] {
3         // Write your code here.
4         return []
5     }
6 }
7
```

Custom OutputRaw OutputSubmit Code

Run or submit code when you're ready.