Our Solution(s) Run Code

Solution 1 Solution 2

```python
# Copyright © 2020 AlgoExpert, LLC. All rights reserved.

# O(w * n * log(n) + n * w * log(w)) time | O(wn) space - where w is the number of
# n is the length of the longest word
def groupAnagrams(words):
    if len(words) == 0:
        return []

    sortedWords = ["".join(sorted(w)) for w in words]
    indices = [i for i in range(len(words))]
    indices.sort(key=lambda x: sortedWords[x])

    result = []
    currentAnagramGroup = []
    currentAnagram = sortedWords[indices[0]]
    for index in indices:
        word = words[index]
        sortedWord = sortedWords[index]

        if sortedWord == currentAnagram:
            currentAnagramGroup.append(word)
            continue

        result.append(currentAnagramGroup)
        currentAnagramGroup = [word]
        currentAnagram = sortedWord

    result.append(currentAnagramGroup)

    return result

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```python
def groupAnagrams(words):
    # Write your code here.
    pass

```

Custom Output Raw Output Submit Code

Run or submit code when you're ready.