

Our Solution(s)

Run Code

Run Code

Solution 1      Solution 2

```

1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <vector>
4 #include <algorithm>
5 #include <unordered_map>
6 using namespace std;
7
8 // O(w * n * log(n)) time | O(wn) space - where w is the number of words and n
9 // is the length of the longest word
10 vector<vector<string>> groupAnagrams(vector<string> words) {
11     unordered_map<string, vector<string>> anagrams;
12
13     for (auto word : words) {
14         string sortedWord = word;
15         sort(sortedWord.begin(), sortedWord.end());
16
17         if (anagrams.find(sortedWord) != anagrams.end()) {
18             anagrams[sortedWord].push_back(word);
19         } else {
20             anagrams[sortedWord] = vector<string>{word};
21         }
22     }
23
24     vector<vector<string>> output = {};
25     for (auto it : anagrams) {
26         output.push_back(it.second);
27     }
28     return output;
29 }
30

```

Your Solutions

Run Code

Run Code

Solution 1    Solution 2    Solution 3

Custom Output   Raw Output   Submit Code

Raw Output [Submit Code](#)

Submit Code

**Run or submit code when you're ready.**