AlgoExpert | Quad Layout | C++ | 14px | Sublime | Monokai | 00:00:0

**Our Solution(s)** Run Code

**Solution 1**

```cpp
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

#include <unordered_map>
#include <vector>
using namespace std;

class TrieNode {
public:
  unordered_map<char, TrieNode *> children;
  string word = "";
};

class Trie {
public:
  TrieNode *root;
  char endSymbol;

  Trie();
  void add(string str);
};

void explore(int i, int j, vector<vector<char>> board, TrieNode *trieN
             vector<vector<bool>> *visited,
             unordered_map<string, bool> *finalWords);
vector<vector<int>> getNeighbors(int i, int j, vector<vector<char>> b

// O(nm*8^s + ws) time | O(nm + ws) space
vector<string> boggleBoard(vector<vector<char>> board, vector<string>
  Trie trie;
  for (string word : words) {
    trie.add(word);
  }
  unordered_map<string, bool> finalWords;
```

**Your Solutions** Run Code

**Solution 1** **Solution 2** **Solution 3**

```cpp
#include <vector>
using namespace std;

vector<string> boggleBoard(vector<vector<char>> board, vector<string>
  // Write your code here.
  return {};
}
```

**Our Tests**

```
[illegible]
```

**Custom Output** Submit Code

```
}

class ProgramTest : public TestSuite {
public:
  void Run() {

    RunTest("Test Case 1", [](){
      vector<vector<char>> board = {
          [illegible]
          [illegible]
          [illegible]
      };
      vector<string> words = {
          [illegible]
          [illegible]
```

Run or submit code when you're ready.