

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 import java.util.*;
4
5 class Program {
6     // O(nm*8^s + ws) time | O(nm + ws) space
7     public static List<String> boggleBoard(char[][] board, String[] words) {
8         Trie trie = new Trie();
9         for (String word : words) {
10             trie.add(word);
11         }
12         Set<String> finalWords = new HashSet<String>();
13         boolean[][] visited = new boolean[board.length][board[0].length];
14         for (int i = 0; i < board.length; i++) {
15             for (int j = 0; j < board[0].length; j++) {
16                 explore(i, j, board, trie.root, visited, finalWords);
17             }
18         }
19         List<String> finalWordsArray = new ArrayList<String>();
20         finalWordsArray.addAll(finalWords);
21         return finalWordsArray;
22     }
23
24     public static void explore(
25         int i,
26         int j,
27         char[][] board,
28         TrieNode trieNode,
29         boolean[][] visited,
30         Set<String> finalWords) {
31         if (visited[i][j]) {
32             return;
33         }
```

Solution 1

Solution 2

Solution 3

```
1 import java.util.*;
2
3 class Program {
4     public static List<String> boggleBoard(char[][] board, String[] words) {
5         // Write your code here.
6         return null;
7     }
8 }
9
```

Our Tests

Custom Output

Submit Code

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

Run or submit code when you're ready.