

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 package main
4
5 // O(nm*8^s + ws) time | O(nm + ws) space
6 func BoggleBoard(board [][]rune, words []string) []string {
7     trie := Trie{children: map[rune]Trie{}}
8     for _, word := range words {
9         trie.Add(word)
10    }
11
12    visited := make([][]bool, len(board))
13    for i := range visited {
14        visited[i] = make([]bool, len(board[i]))
15    }
16
17    finalWords := map[string]bool{}
18    for i := range board {
19        for j := range board[i] {
20            explore(i, j, board, trie, visited, finalWords)
21        }
22    }
23
24    result := []string{}
25    for word := range finalWords {
26        result = append(result, word)
27    }
28    return result
29 }
30
31 func explore(i, j int, board [][]rune, trie Trie, visited [][]bool, f:
32     if visited[i][j] {
33         return
```

Solution 1

Solution 2

Solution 3

```
1 package main
2
3 func BoggleBoard(board [][]rune, words []string) []string {
4     // Write your code here.
5     return nil
6 }
7
```

Our Tests

Custom Output

Submit Code

```
1 package main
2
3 func BoggleBoard(board [][]rune, words []string) []string {
4     // Write your code here.
5     return nil
6 }
```

```

17 # Run on Hadoop Distributed Filesystem (HDFS)
18 input = ["input"]
19
20 # List of input files
21 input_files = ["input1", "input2", "input3", "input4", "input5", "input6", "input7", "input8", "input9", "input10"]
22
23 # List of output files
24 output_files = ["output1", "output2", "output3", "output4", "output5", "output6", "output7", "output8", "output9", "output10"]
25
26 # MapReduce job
27 job = MapReduceJob(input_files, output_files)
28
29 # Run the job
30 job.run()
31
32 # Print the output
33 for output_file in output_files:
34     print(output_file)
35
36 # End of script
37

```

Run or submit code when you're ready.