

PromptScratchpadOur Solution(s)Video Explanation

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, finalWords map[string]bool) {
32     if visited[i][j] {
33         return
34     }
35     letter := board[i][j]
36     if _, found := trie.children[letter]; !found {
37         return
38     }
39     visited[i][j] = true
40     trie = trie.children[letter]
41     if end, found := trie.children['*']; found {
42         finalWords[end.word] = true
43     }
44     neighbors := getNeighbors(i, j, board)
45     for _, neighbor := range neighbors {
46         explore(neighbor[0], neighbor[1], board, trie, visited, finalWords)
47     }
48     visited[i][j] = false
49 }
50
51 func getNeighbors(i, j int, board [][]rune) [][]int {
52     neighbors := [][]int{}
53     if i > 0 && j > 0 {
54         neighbors = append(neighbors, []int{i - 1, j - 1})
55     }
56     if i > 0 && j < len(board[0])-1 {
57         neighbors = append(neighbors, []int{i - 1, j + 1})
58     }
59     if i < len(board)-1 && j < len(board[0])-1 {
60         neighbors = append(neighbors, []int{i + 1, j + 1})
61     }
62     if i < len(board)-1 && j > 0 {
63         neighbors = append(neighbors, []int{i + 1, j - 1})
64     }
65     if i > 0 {
66         neighbors = append(neighbors, []int{i - 1, j})
67     }
68     if i < len(board)-1 {
69         neighbors = append(neighbors, []int{i + 1, j})
70     }
71     if j > 0 {
72         neighbors = append(neighbors, []int{i, j - 1})
73     }
74     if j < len(board[0])-1 {
75         neighbors = append(neighbors, []int{i, j + 1})
76     }
77     return neighbors
78 }
79
80
81 type Trie struct {
82     children map[rune]Trie
83
84     word string
85 }
86
```