

PromptScratchpadOur Solution(s)Video Explanation

Run Code

Solution 1

```
39         let nextNode = trieNode.children[letter] as! TrieNode
40
41         if nextNode.children.keys.contains("*") {
42             if let word = nextNode.children["*"] as? String {
43                 finalWords[word] = true
44             }
45         }
46
47         let neighbors = getNeighbors(i, j, board: board)
48
49         for neighbor in neighbors {
50             explore(neighbor[0], neighbor[1], board, nextNode, &visited, &finalWords)
51         }
52
53         visited[i][j] = false
54     }
55
56     func getNeighbors(_ i: Int, _ j: Int, board: [[String]]) -> [[Int]] {
57         var neighbors = [[Int]]()
58
59         if i > 0, j > 0 {
60             neighbors.append([i - 1, j - 1])
61         }
62
63         if i > 0 {
64             neighbors.append([i - 1, j])
65         }
66
67         if i > 0, j < board[i].count - 1 {
68             neighbors.append([i - 1, j + 1])
69         }
70
71         if j < board[i].count - 1 {
72             neighbors.append([i, j + 1])
73         }
74
75         if i < board.count - 1, j < board[i].count - 1 {
76             neighbors.append([i + 1, j + 1])
77         }
78
79         if i < board.count - 1 {
80             neighbors.append([i + 1, j])
81         }
82
83         if i < board.count - 1, j > 0 {
84             neighbors.append([i + 1, j - 1])
85         }
86
87         if j > 0 {
88             neighbors.append([i, j - 1])
89         }
90
91         return neighbors
92     }
93
94     class TrieNode {
95         var children: [String: Any] = [:]
96     }
97
98     class Trie {
99         var root: TrieNode
100         let endSymbol: String = "*"
101
102         init() {
103             root = TrieNode()
104         }
105
106         func add(_ word: String) {
107             var current = root
108
109             for character in word {
110                 let stringifiedCharacter = String(character)
111
112                 if !current.children.keys.contains(stringifiedCharacter) {
113                     current.children[stringifiedCharacter] = TrieNode()
114                 }
115
116                 let nextNode = current.children[stringifiedCharacter] as! TrieNode
117                 current = nextNode
118             }
119
120             current.children[endSymbol] = word
121         }
122     }
123 }
124
```