Sublime Monokai 00:00:00 AlgoExpert **Quad Layout** JavaScript 12px

Prompt Run Code

Scratchpad

Our Solution(s)

Solution 1

Video Explanation

```
1\, // Copyright @ 2020 AlgoExpert, LLC. All rights reserved.
 3 \ // \ O(nm*8^s + ws) \ time \ | \ O(nm + ws) \ space
 4 function boggleBoard(board, words) {
      const trie = new Trie();
      for (const word of words) \{
        trie.add(word);
 8
      const finalWords = {};
10
      const visited = board.map(row => row.map(letter => false));
11
      for (let i = 0; i < board.length; i++) {
        for (let j = 0; j < board[i].length; j++) {</pre>
           explore(i, j, board, trie.root, visited, finalWords);
13
14
15
      return Object.keys(finalWords);
16
17 }
18
19
    function explore(i, j, board, trieNode, visited, finalWords) {
20
      if (visited[i][j]) return;
      const letter = board[i][j];
21
      if (!(letter in trieNode)) return;
22
      visited[i][j] = true;
24
      trieNode = trieNode[letter];
25
      if ('*' in trieNode) finalWords[trieNode['*']] = true;
      const neighbors = getNeighbors(i, j, board);
26
27
      for (const neighbor of neighbors) {
28
        explore(neighbor[0], neighbor[1], board, trieNode, visited, finalWords);
29
30
      visited[i][j] = false;
31 }
32
    function getNeighbors(i, j, board) {
33
      const neighbors = [];
      if (i > 0 && j > 0) {
35
36
        neighbors.push([i - 1, j - 1]);
37
       if \ (i \ > \ 0 \ \&\& \ j \ < \ board[0].length \ - \ 1) \ \{ \\
38
39
        neighbors.push([i - 1, j + 1]);
40
       \mbox{if (i < board.length - 1 \&\& j < board[0].length - 1) } \{ \\
41
42
        neighbors.push([i + 1, j + 1]);
43
44
       \  \  \text{if} \ (\texttt{i} \ < \ \text{board.length} \ - \ 1 \ \&\& \ \texttt{j} \ > \ \texttt{0}) \ \{ \\
45
        neighbors.push([i + 1, j - 1]);
46
47
      \quad \textbf{if} \ (\texttt{i} \ \gt \ \texttt{0}) \ \{
        neighbors.push([i - 1, j]);
48
49
50
      if (i < board.length - 1) {</pre>
        neighbors.push([i + 1, j]);
51
52
53
      if (j > 0) {
54
        neighbors.push([i, j - 1]);
55
56
      if (j < board[0].length - 1) {
57
        neighbors.push([i, j + 1]);
58
59
      return neighbors;
60
61
62 class Trie {
63
64
        this.root = {};
        this.endSymbol = '*';
65
66
67
68
      \mathsf{add}(\mathsf{word}) \ \{
69
        let current = this.root;
70
         for (const letter of word) \{
          if (!(letter in current)) current[letter] = {};
71
72
           current = current[letter];
73
74
         current[this.endSymbol] = word;
75
76 }
78 exports.boggleBoard = boggleBoard;
```