AlgoExpert

Solution 1

4

9

10

11

12

13

14 15

17

18

19 20

21

22

23

24

25

26 27

28

29

30 31

33

Quad Layout

Python

Sublime

Monokai

00:00:

Run Code

Our Solution(s)

def boggleBoard(board, words):

for i in range(len(board)):

if letter not in trieNode:

trieNode = trieNode[letter]

 $\quad \text{for neighbor in neighbors:} \quad$

visited[i][j] = False

32 def getNeighbors(i, j, board):

neighbors = []

finalWords[trieNode["*"]] = True
neighbors = getNeighbors(i, j, board)

return list(finalWords.keys())

for j in range(len(board[i])):

16 def explore(i, j, board, trieNode, visited, finalWords):

trie = Trie()
for word in words:
 trie.add(word)
finalWords = {}

if visited[i][j]:

return
letter = board[i][j]

return

visited[i][j] = True

if "*" in trieNode:

1 # Copyright © 2020 AlgoExpert, LLC. All rights reserved.

visited = [[False for letter in row] for row in board]

explore(i, j, board, trie.root, visited, finalWords)

explore(neighbor[0], neighbor[1], board, trieNode, visited, fi

O(nm*8^s + ws) time | O(nm + ws) space

Run Code

Your Solutions

14рх

Solution 1 Solution 2 Solution 3

```
def boggleBoard(board, words):
    # Write your code here.

pass
4
```

Our Tests

Custom Output

Submit Code

Run or submit code when you're ready.