

361. Bomb Enemy

Medium

695

89

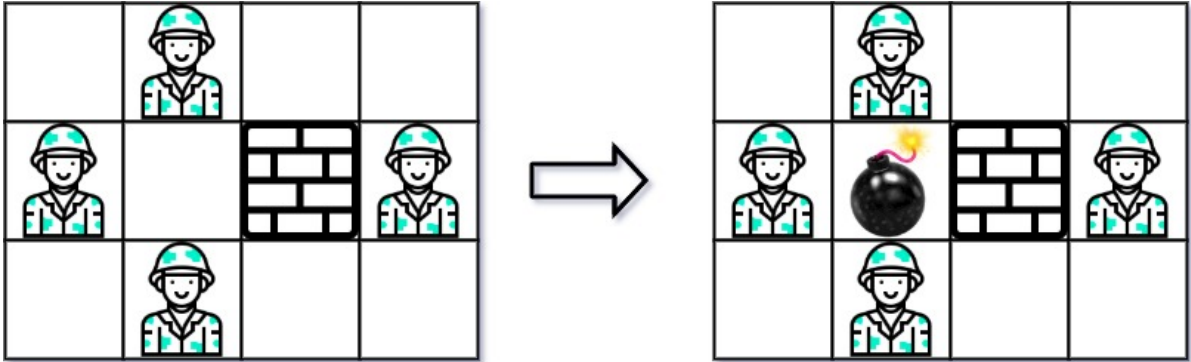
Add to List

Share

Given an $m \times n$ matrix `grid` where each cell is either a wall `'W'`, an enemy `'E'` or empty `'0'`, return the *maximum enemies you can kill using one bomb*. You can only place the bomb in an empty cell.

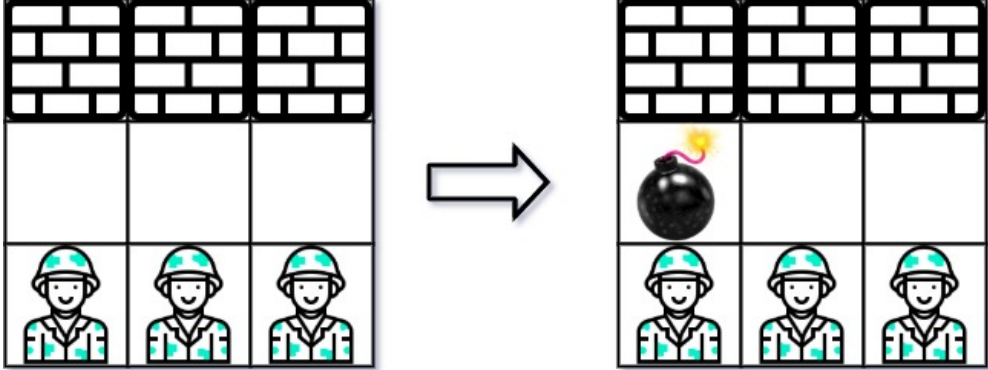
The bomb kills all the enemies in the same row and column from the planted point until it hits the wall since it is too strong to be destroyed.

Example 1:



Input: `grid = [["0","E","0","0"], ["E","0","W","E"], ["0","E","0","0"], ["0","0","0","0"]]`
Output: 3

Example 2:



Input: `grid = [["W","W","W","W"], ["0","0","0","0"], ["E","E","E","E"], ["0","0","0","0"]]`
Output: 1

Constraints:

- $m == \text{grid.length}$
- $n == \text{grid}[i].length$
- $1 \leq m, n \leq 500$
- `grid[i][j]` is either `'W'`, `'E'`, or `'0'`.

Accepted 60,209 | Submissions 122,500

Seen this question in a real interview before?

Yes

No

Companies

1

0 - 6 months

6 months - 1 year

1 year - 2 years

LinkedIn

5

Related Topics

Array

Dynamic Programming

Matrix

```
1 class Solution {
2     public int maxKilledEnemies(char[][] grid) {
3         if (grid.length == 0)
4             return 0;
5
6         int rows = grid.length;
7         int cols = grid[0].length;
8
9         int maxCount = 0, rowHits = 0;
10        int[] colHits = new int[cols];
11
12        for (int row = 0; row < rows; ++row) {
13            for (int col = 0; col < cols; ++col) {
14
15                // reset the hits on the row, if necessary.
16                if (col == 0 || grid[row][col - 1] == 'W') {
17                    rowHits = 0;
18                    for (int k = col; k < cols; ++k) {
19                        if (grid[row][k] == 'W')
20                            // stop the scan when we hit the wall.
21                            break;
22                        else if (grid[row][k] == 'E')
23                            rowHits += 1;
24                    }
25                }
26
27                // reset the hits on the column, if necessary.
28                if (row == 0 || grid[row - 1][col] == 'W') {
29                    colHits[col] = 0;
30                    for (int k = row; k < rows; ++k) {
31                        if (grid[k][col] == 'W')
32                            break;
33                        else if (grid[k][col] == 'E')
34                            colHits[col] += 1;
35                    }
36                }
37
38                // run the calculation for the empty cell.
39                if (grid[row][col] == '0') {
40                    maxCount = Math.max(maxCount, rowHits + colHits[col]);
41                }
42            }
43        }
44        return maxCount;
45    }
46 }
47 }
```