

Advanced Programming Practices

Hackerrank Week-10

Name: Ronit Kumar

Reg: RA2111032010009

Section: T2

Branch:CSE w/s OT

Ques 1: The Bomberman Game

Problem:

The screenshot shows the HackerRank interface for the 'The Bomberman Game' problem. At the top, there's a navigation bar with 'HackerRank', 'Prepare', 'Certify', and 'Compete' tabs. Below this, the problem title 'The Bomberman Game' is displayed with a star icon. A progress bar indicates '184 more points to get your next star!' with a rank of 651360 and 231/475 points. A notification box states: 'Your The Bomberman Game submission got 40.00 points. You are now 184 points away from the 4th star for your problem solving badge. Try the next challenge Try a Random Challenge'. The main content area is divided into 'Problem', 'Submissions', 'Leaderboard', 'Discussions', and 'Editorial' tabs. The 'Problem' tab is active, showing the problem description. The sidebar on the right contains metadata: Author (kevinsogo), Difficulty (Medium), Max Score (40), Submitted By (28980), and a 'NEED HELP?' section with links to 'View discussions', 'View editorial', and 'View top submissions'. Below this is a 'RATE THIS CHALLENGE' section with five stars and a 'MORE DETAILS' section with links to 'Download problem statement', 'Download sample test cases', and 'Suggest Edits'.

Problem

Bomberman lives in a rectangular grid. Each cell in the grid either contains a bomb or nothing at all. Each bomb can be planted in any cell of the grid but once planted, it will detonate after exactly 3 seconds. Once a bomb detonates, it's destroyed — along with anything in its four neighboring cells. This means that if a bomb detonates in cell (i, j) any valid cells $(i \pm 1, j)$ and $(i, j \pm 1)$ are cleared. If there is a bomb in a neighboring cell, the neighboring bomb is destroyed without detonating, so there's no chain reaction.

Bomberman is immune to bombs, so he can move freely throughout the grid. Here's what he does:

1. Initially, Bomberman arbitrarily plants bombs in some of the cells, the initial state.
2. After one second, Bomberman does nothing.
3. After one more second, Bomberman plants bombs in all cells without bombs, thus filling the whole grid with bombs. No bombs detonate at this point.
4. After one more second, any bombs planted exactly three seconds ago will detonate. Here, Bomberman stands back and observes.
5. Bomberman then repeats steps 3 and 4 indefinitely.

Note that during every second Bomberman plants bombs, the bombs are planted simultaneously (i.e., at the exact same moment), and any bombs planted at the same time will detonate at the same time.

Given the initial configuration of the grid with the locations of Bomberman's first batch of planted bombs, determine the state of the grid after N seconds.

For example, if the initial grid looks like:

```
...  
.0.  
.
```

Code:

```
def bomb(b,r,c):  
    field = [['0' for i in range(c)] for j in  
range(r)]  
    for i in range(r):  
        for j in range(c):  
            if b[i][j] == '0':  
                field[i][j] = '.'  
                if i+1<r:  
                    field[i+1][j] = '.'  
                if i>0:  
                    field[i-1][j] = '.'
```

```

        if j+1<c:
            field[i][j+1] = '.'
        if j>0:
            field[i][j-1] = '.'

    return field

r,c,n = input().split()
r,c,n = int(r),int(c),int(n)
b = []
for i in range(r):
    row = list(input())
    b.append(row)
if n%2==0:
    f = [['0' for i in range(c)] for j in range(r)]
    for i in range(r):
        print(''.join(map(str,f[i])))
else:
    bombed1 = bomb(b,r,c)
    bombed2 = bomb(bombed1,r,c)

    if n==1:
        for i in range(r):
            print(''.join(map(str,b[i])))
    elif (n+1)%4==0:
        for i in range(r):
            print(''.join(map(str,bombed1[i])))
    elif (n+2)%4==0:
        for i in range(r):

```

```

        print(''.join(map(str,b[i])))
    else:
        for i in range(r):
            print(''.join(map(str,bombed2[i])))
)

```

Output:

The screenshot shows a web browser window with the HackerRank website. The top navigation bar includes links to Instagram, Chats, YouTube, and various HackerRank-related pages. The main content area displays the code for 'The Bomberman Game' challenge, which is a Python script. The code is as follows:

```

37:         print(''.join(map(str,b[i])))
38:     else:
39:         for i in range(r):
40:             print(''.join(map(str,bombed2[i])))
41: )
42:

```

Below the code editor, there are buttons for 'Upload Code as File', 'Test against custom input', 'Run Code', and 'Submit Code'. The 'Submit Code' button is highlighted in green, indicating a successful submission.

A green banner with the text 'Congratulations' and 'You solved this challenge. Would you like to challenge your friends?' is displayed. Below this, there are social media sharing icons for Facebook, Twitter, and LinkedIn, and a 'Next Challenge' button.

The 'Test case 0' section shows the compiler message 'Success' and the input (stdin) for the test case:

```

1 6 7 3
2 .....
3 ...0...
4 ...0...
5 .....
6 00.....
7 00.....

```

The bottom of the screen shows a dock with various application icons, including Finder, Safari, Mail, App Store, Photos, Music, and others.