5202. Largest Magic Square

My Submissions (/contest/biweekly-contest-54/problems/largest-magic-square/submissions/)

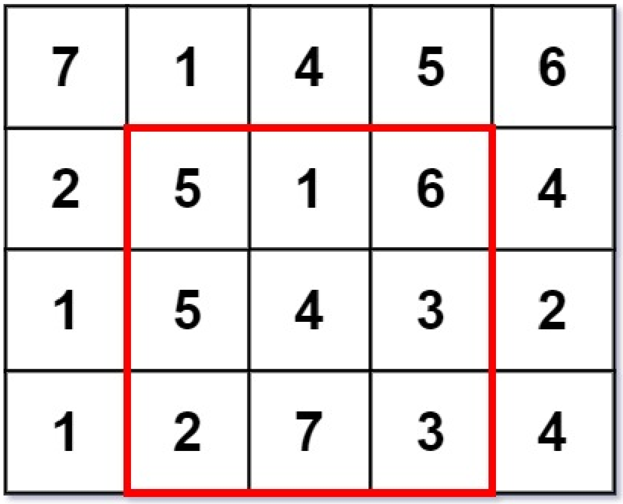
Back to Contest (/contest/biweekly-contest-54/)

User Accepted:	1759
User Tried:	2355
Total Accepted:	1806
Total Submissions:	4598
Difficulty:	Medium

A `k x k` **magic square** is a `k x k` grid filled with integers such that every row sum, every column sum, and both diagonal sums are **all equal**. The integers in the magic square **do not have to be distinct**. Every `1 x 1` grid is trivially a **magic square**.

Given an `m x n` integer `grid`, return *the **size** (i.e., the side length* `k`*) of the **largest magic square** that can be found within this grid*.

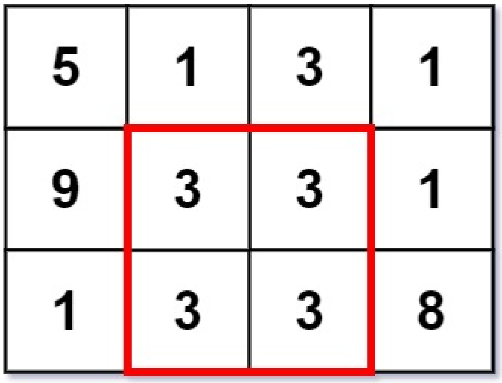
Example 1:



7	1	4	5	6
2	5	1	6	4
1	5	4	3	2
1	2	7	3	4

```
Input: grid = [[7,1,4,5,6],[2,5,1,6,4],[1,5,4,3,2],[1,2,7,3,4]]
Output: 3
Explanation: The largest magic square has a size of 3.
Every row sum, column sum, and diagonal sum of this magic square is equal to 12.
- Row sums: 5+1+6 = 5+4+3 = 2+7+3 = 12
- Column sums: 5+5+2 = 1+4+7 = 6+3+3 = 12
- Diagonal sums: 5+4+3 = 6+4+2 = 12
```

Example 2:



5	1	3	1
9	3	3	1
1	3	3	8

```
Input: grid = [[5,1,3,1],[9,3,3,1],[1,3,3,8]]
Output: 2
```

Constraints:

- m == grid.length
- n == grid[i].length
- 1 <= m, n <= 50
- 1 <= grid[i][j] <= 10^6

JavaScript

```
const pr = console.log;
const largestMagicSquare = (g) => {
    let n = g.length;
    let m = g[0].length;
    for (let edge = Math.min(n, m); edge > 1; edge--) {
        for (let i = 0; i < n; i++) {
            for (let j = 0; j < m; j++) {
                let nexti = i + edge - 1;
                let nextj = j + edge - 1;
                if (nexti >= n || nextj >= m) continue;
                let topL = g[i][j];
                let bottomL = g[nexti][j];
                let topR = g[i][nextj];
                let bottomR = g[nexti][nextj];
                pr(topL, bottomL, topR, bottomR, "edge", edge)
                if (ok(g, i, j, nexti, nextj)) return edge;
            }
        }
    }
    return 1;
};

const ok = (g, i, j, nexti, nextj) => {
    let rowse = new Set();
    for (let row = i; row <= nexti; row++) {
        let rowSum = 0;
        for (let col = j; col <= nextj; col++) {
            rowSum += g[row][col];
        }
        rowse.add(rowSum);
        if (rowse.size > 1) return false;
    }
    let colse = new Set();
    for (let col = j; col <= nextj; col++) {
        let colSum = 0;
        for (let row = i; row <= nexti; row++) {
            colSum += g[row][col];
        }
        colse.add(colSum);
        if (colse.size > 1) return false;
    }
    pr(rowse, colse);
    if (rowse.values().next().value != colse.values().next().value) return false;
    let [x, y] = [i, j];
    let d1sum = g[x][y];
    while (x + 1 <= nexti && y + 1 <= nextj) {
        x++;
        y++;
        d1sum += g[x][y];
        // pr("d1sum", d1sum);
    }
    pr("d1sum", d1sum);
    x = nexti;
    y = j;
    let d2sum = g[x][y];
    while (x - 1 >= i && y + 1 <= nextj) {
        x--;
        y++;
        d2sum += g[x][y];
    }
    pr("d2sum", d2sum);
    if (rowse.values().next().value != d1sum) return false;
    if (d1sum != d2sum) return false;
    return true;
};
```

□ **Custom Testcase** Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/506878905/) More Details (/submissions/detail/506878905/)

Share your acceptance!

1

Help Center (/support) | Jobs (/jobs) | Bug Bounty (/bugbounty) | Online Interview (/interview/) | Students (/student) | Terms (/terms) | Privacy Policy (/privacy)

United States (/region)