

6433. Maximum Number of Moves in a Grid

My Submissions (/contest/weekly-contest-345/problems/maximum-number-of-moves-in-a-grid/submissions/)

Back to Contest (/contest/weekly-contest-345/)

You are given a **0-indexed** $m \times n$ matrix **grid** consisting of **positive** integers.

You can start at **any** cell in the first column of the matrix, and traverse the grid in the following way:

- From a cell (row, col) , you can move to any of the cells: $(row - 1, col + 1)$, $(row, col + 1)$ and $(row + 1, col + 1)$ such that the value of the cell you move to, should be **strictly** bigger than the value of the current cell.

Return the **maximum** number of **moves** that you can perform.

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Medium

Example 1:

2	4	3	5
5	4	9	3
3	4	2	11
10	9	13	15

Input: grid = [[2,4,3,5],[5,4,9,3],[3,4,2,11],[10,9,13,15]]
Output: 3
Explanation: We can start at the cell (0, 0) and make the following moves:
- (0, 0) -> (0, 1).
- (0, 1) -> (1, 2).
- (1, 2) -> (2, 3).
It can be shown that it is the maximum number of moves that can be made.

Example 2:

3	2	4
2	1	9
1	1	7

Input: grid = [[3,2,4],[2,1,9],[1,1,7]]
Output: 0
Explanation: Starting from any cell in the first column we cannot perform any moves.

Constraints:

- $m == \text{grid.length}$
- $n == \text{grid}[i].length$
- $2 \leq m, n \leq 1000$
- $4 \leq m * n \leq 10^5$
- $1 \leq \text{grid}[i][j] \leq 10^6$

JavaScript

📄

↺

⚙️

```
1  const initialize2DArray = (n, m) => [...Array(n)].map(() => Array(m).fill(0));
2
3  const maxMoves = (g) => {
4      let n = g.length, m = g[0].length, f = initialize2DArray(n, m);
5      for (let i = 0; i < n; i++) f[i][0] = 1;
6      for (let j = 1; j < m; j++) {
7          for (let i = 0; i < n; i++) {
8              // for (let i = 0; i < n; i++) {
9              //     for (let j = 1; j < m; j++) {
```


```
10 ▾      if (g[i][j] > g[i][j - 1] && f[i][j - 1]) {
11          f[i][j] = 1;
12      }
13 ▾      if (i - 1 >= 0 && g[i][j] > g[i - 1][j - 1] && f[i - 1][j - 1]) {
14          f[i][j] = 1;
15      }
16 ▾      if (i + 1 < n && g[i][j] > g[i + 1][j - 1] && f[i + 1][j - 1]) {
17          f[i][j] = 1;
18      }
19      }
20  }
21 ▾  for (let j = m - 1; j >= 0; j--) {
22      let canReach = false;
23 ▾      for (let i = 0; i < n; i++) {
24 ▾          if (f[i][j]) {
25              canReach = true;
26              break;
27          }
28      }
29      if (canReach) return j;
30  }
31  return 0;
32  };
```

☐ Custom Testcase[Use Example Testcases](#)[Run](#)[Submit](#)Submission Result: **Accepted** (/submissions/detail/950008401/) ?[More Details](#) (/submissions/detail/950008401/)

Share your acceptance!

[◀2](#)

Copyright © 2023 LeetCode

[Help Center](#) (/support) | [Jobs](#) (/jobs) | [Bug Bounty](#) (/bugbounty) | [Online Interview](#) (/interview/) | [Students](#) (/student) | [Terms](#) (/terms) | [Privacy Policy](#) (/privacy) [United States](#) (/region)