Largest Square of Zeros

Presented by
Soe Min Min Latt - 6611938
Win Yu Maung - 6612054

CSX3009 - 541

Table Of Contents

PROBLEM DESCRIPTION

Given a two-dimensional matrix of size H × W consisting only of binary values 0 and 1, the task is to determine the largest square submatrix that contains only 0s.

The input starts with two integers H and W, representing the height and width of the matrix. This is followed by H rows, each containing W elements (0 or 1).

The objective is to find the area (number of cells) of the largest possible square formed entirely by 0s within the matrix.

SOURCE

Website : https://onlinejudge.u-aizu.ac.jp/

Problem Name : Largest Square

Problem Code : DPL_3_A

Largest Square

Given a matrix ($H \times W$) which contains only 1 and 0, find the area of the largest square matrix which only contains 0s.

Input

$$
\begin{array}{l}
H \ W \\
c_{1,1} \ c_{1,2} \ \dots \ c_{1,W} \\
c_{2,1} \ c_{2,2} \ \dots \ c_{2,W} \\
\vdots \\
c_{H,1} \ c_{H,2} \ \dots \ c_{H,W}
\end{array}
$$

In the first line, two integers H and W separated by a space character are given. In the following H lines, $c_{i,j}$, elements of the $H \times W$ matrix, are given.

Output

Print the area (the number of 0s) of the largest square.

Constraints

- $1 \le H, W \le 1{,}400$

Sample Input

```
4 5
0 0 1 0 0
1 0 0 0 0
0 0 0 1 0
0 0 0 1 0
```

Sample Output

```
4
```

WHY THIS PROBLEM

- We chose this problem because it is a classic dynamic programming problem that strengthens problem-solving skills.
- It efficiently handles large inputs, helping me learn optimization and space management.
- The problem has a clear real-world relevance in image processing and pattern recognition.

INPUT FORMAT

- Two integers H and W (height and width)
- H rows of W elements (0 or 1)

Sample input:

H W	4 5
$c_{1,1}$ $c_{1,2}$ ... $c_{1,W}$	0 0 1 0 0
$c_{2,1}$ $c_{2,2}$ ... $c_{2,W}$	1 0 0 0 0
:	0 0 0 1 0
$c_{H,1}$ $c_{H,2}$ ... $c_{H,W}$	0 0 0 1 0

Problem Constraint: $1 \le H, W \le 1{,}400$

OUTPUT FORMAT

- An integer representing the area of the largest square containing only 0s
- Print the area (the number of 0s) of the largest square.

Sample output: 4

KEY OBSERVATION

- Instead of checking all squares again and again, we:
  - Look at the matrix one cell at a time
  - Build bigger squares using smaller squares
- Why does this work?
  - A square can grow only if the cell above, left, and top-left are all 0
  - So we just remember the biggest square possible at each cell, which makes the solution fast and simple

ALGORITHM OVERVIEW

APPROACH: DYNAMIC PROGRAMMING

- Create a 2D DP array where dp[i][j] = size of the largest square of 0s ending at cell (i, j)
- Base idea:
  - If the cell value is 1, no square can end there → dp[i][j] = 0
  - If the cell value is 0, we try to form a square
- For each cell (i, j) with value 0:
  - Look at three neighboring cells:
    - Top: dp[i-1][j]
    - Left: dp[i][j-1]
    - Top-left: dp[i-1][j-1]
  - Set
    - dp[i][j] = min(top, left, top-left) + 1
- Keep track of the maximum square size found during computation
- Final answer:
  - Area of the largest square = (maximum size) × (maximum size)

COMPLEXITY

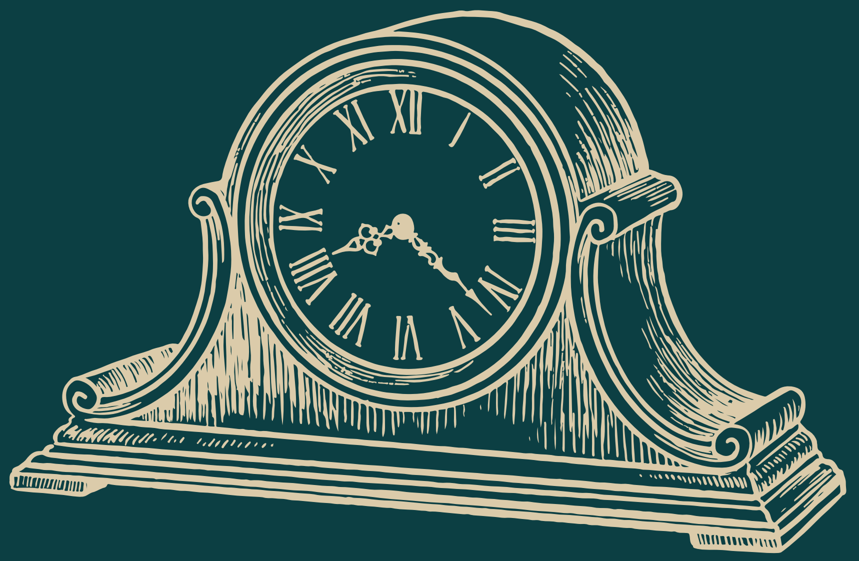
Time Complexity: O(H × W)

Space Complexity: O(H × W)

where

H = number of rows in the matrix (height)

W = number of columns in the matrix (width)



THANK YOU