

LC-1277

Page No.:

Date:

youva

Count all submatrices with all ones
(square)

★ The approach followed here will be similar to LC221 - Maximal Square, wherein we had to find the area of maximal square of 1's in the matrix.

In that question, we used a Tabulation DP approach. At every index (i, j) of the DP matrix, we stored the maximum side length of square that can be formed to the top left of it.



5 denotes that upto this cell, a max. of square side of 5 units can be formed.

In this question, our approach will be similar.

1	0	1	0	0
1	0	1	1	1
1	1	1	1	1
1	0	0	1	0

→ All the 1's at boundary, will obviously make a square of size 1 each. So $result += 1$ for each such 1 at boundary.

→ We will skip all 0s.

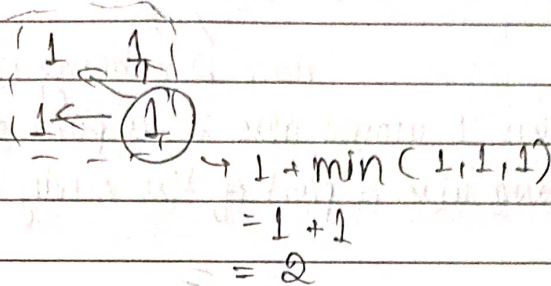
→ For every 1, we will see the

$result += 1 + \min(\overset{\text{top}}{\text{grid}[i-1][j]}, \overset{\text{left}}{\text{grid}[i][j-1]}, \underset{\text{top-left}}{\text{grid}[i-1][j-1]})$

→ The idea is that since we know that

$1 + \min(\text{grid}[i-1][j], \text{grid}[i][j-1], \text{grid}[i-1][j-1])$
is the max. length of square that can also be
formed using the current 1.

Say,

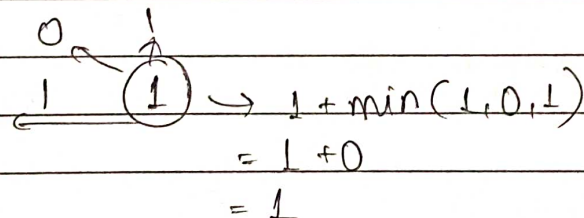


Denotes that 1 square is made alone by this 1.

and 1 square of side 2 units is made with the
previous 1s.

so in total $1 + 1 = 2$ squares can be made from
this current 1.

Say,



There is no way of expanding the previous square of 1's
using this current square, since a 0 exists. so only 1
square made using the current '1' is considered.

→ CODE:-

```

int squares = 0;
for (int i = 0; i < grid.size(); i++)
    for (int j = 0; j < grid[0].size(); j++)
    {
        if (grid[i][j] == 0) continue;
        else squares += (grid[i][j] = 1 + min(grid[i-1][j],
                                                grid[i][j-1], grid[i-1][j-1]));
    }
return squares;

```

→ example:

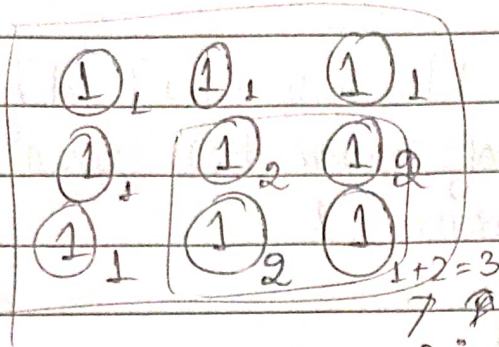
Here,

①

← old value

2 ← updated value

$\Rightarrow 1 + \min(\text{top, left, top-left})$



Here 2 is being added because this 1 would also be a part of 2×2 square and also a part of the newly discovered 3×3 square.