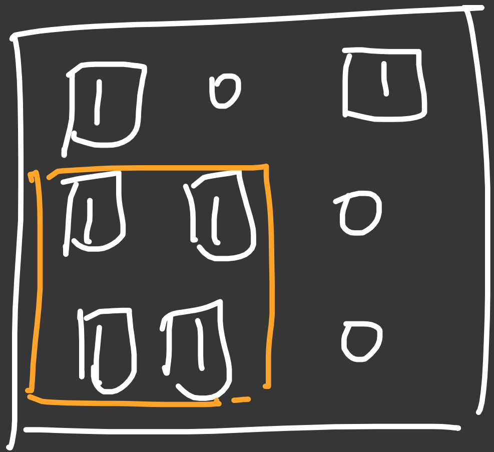
Count No. of Square Submatrix



7 square

1 size → 6 sq

2 size → ~~4~~ 1 sq.

7 sq

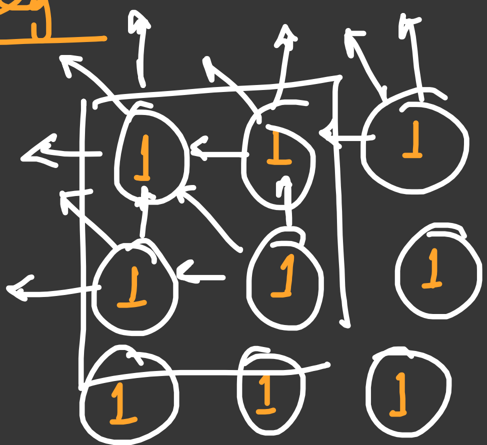
Brute force:

- We stand at every index at count the no of square it can be by checking side left right and diagonal

DP approach:

- Idea is to use dp array/matrix to store the no. of square a index can make and use dp array to fill further dp array by ~~p~~ up, left and diagonal.

eg.



$$\longrightarrow \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 2 & 3 \end{bmatrix}$$

How we dro fill the dp matrax?

- We taking the minimal of the up, left, and left diagonal of the dp array
- 1st row and col is same in dp-array as in matrix

eg:

1	1	0		1	1	0
1	1	1	$\longrightarrow$	1	2	1
1	1	0		1	2	2

ans would be soum of all the element in matrix.

Pseudo code.

dp[n][m] = 0

Base Case:

```
for ( i = 0; i < n; i++)
    dp [i][0] = arr[i][0];

for ( i = 0, i < m, i++)
    dp[0][i] = arr[0][i];
```

Main loop/filling dp

```
for ( i = 1, i < n; i++)
```

```
for( j=1, j<n; j++)
    dp[i][j] = min( dp[i-1][j],
                    dp[i][j-1],
                    dp[i-1][j-1])
               + 1

for (i=0, i<n, i++)
    for( j = 0, j<m, j++)
        sum += dp[i][j],
return sum;
```

T. C. $O(N) + O(M) + O(N \times M) + O(N \times M)$

$\approx O(N \times M)$

S.C. : $O(N \times M) \longrightarrow dp$