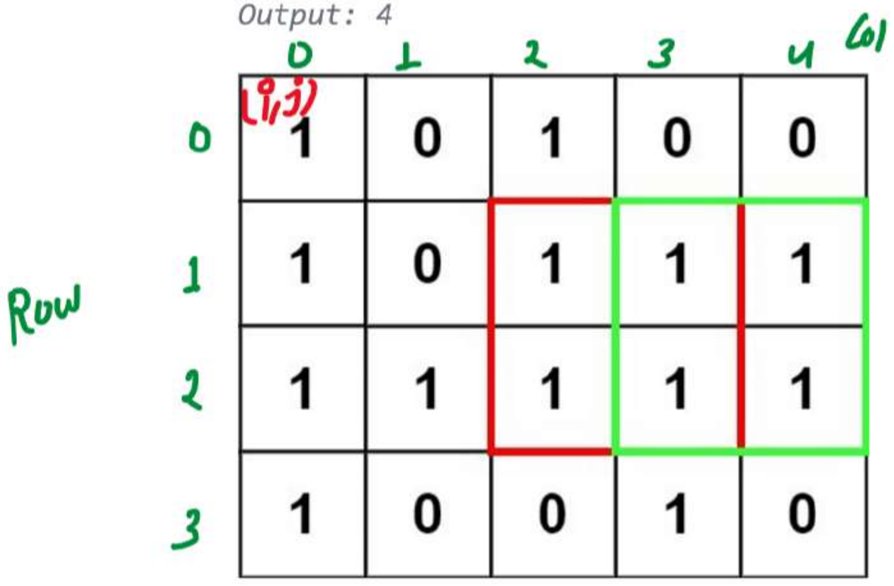
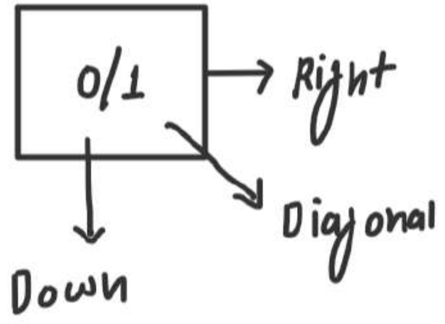
6. Maximal Square (Leetcode-221)

```
Problem Statement:
Given an m x n binary matrix filled with 0's and 1's, find the largest square containing only 1's
and return its area.

Input: matrix =
[["1","0","1","0","0"],["1","0","1","1","1"],["1","1","1","1","1"],["1","0","0","1","0"]]
Output: 4
```





ANS

$\min(Ri, Di, Do) + \boxed{1}$

Maxi

$\max(ANS, maxi)$

Area

$maxi * maxi$

Row=5

Col = 4

max(1, 2)

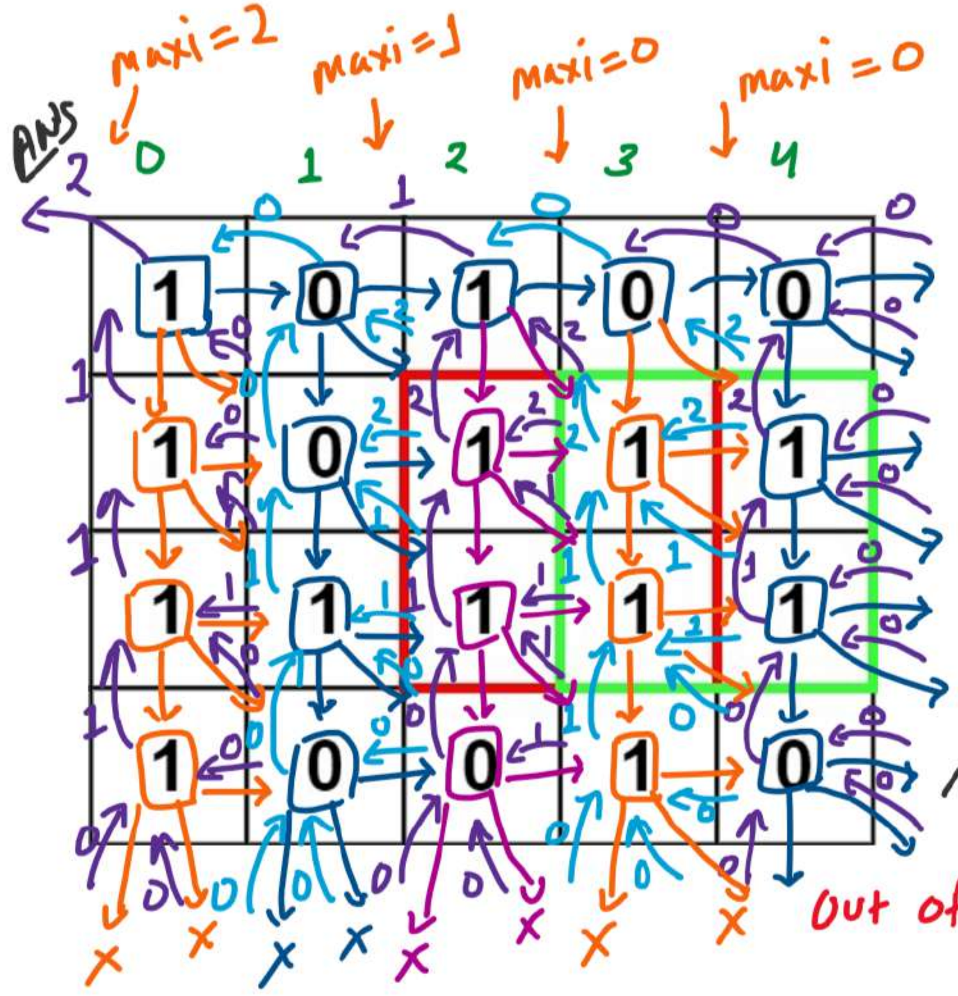
maxi = max(maxi, Ans)

= 2

Area = 2x2

= $\boxed{4}$

Output



Out of Bound

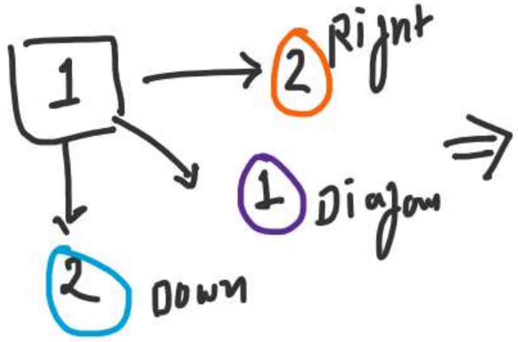
Base case
(i >= Row)
return 0

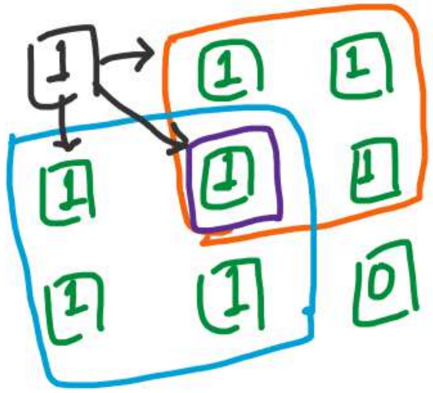
ANS

min(Ri, Di, DO) + $\boxed{1}$

Base case
(j >= col)
return 0

Out of Bound

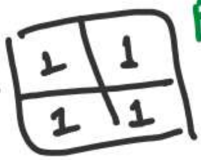




ANS = $\boxed{1}$ + Min(2, 1, 2)

= 1 + 1

= 2 →



Right square because No zero contained ✓

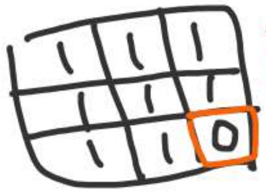
Area = 4

ANS = $\boxed{1}$ + max(2, 1, 2)

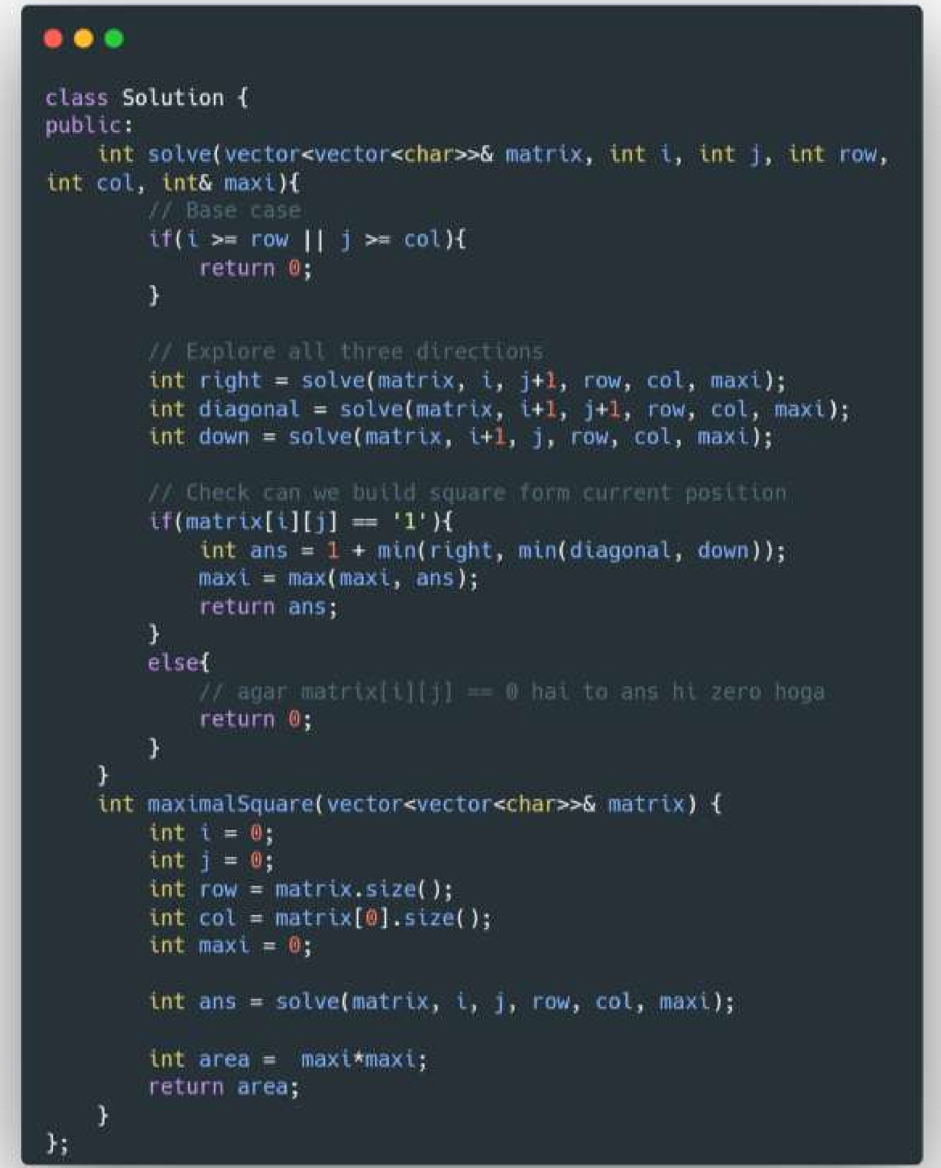
= $\boxed{1}$ + 2

= 3 →

Area = 9



X WRONG Square because of zero contained



```cpp
class Solution {
public:
    int solve(vector<vector<char>>& matrix, int i, int j, int row,
int col, int& maxi){
        // Base case
        if(i >= row || j >= col){
            return 0;
        }

        // Explore all three directions
        int right = solve(matrix, i, j+1, row, col, maxi);
        int diagonal = solve(matrix, i+1, j+1, row, col, maxi);
        int down = solve(matrix, i+1, j, row, col, maxi);

        // Check can we build square form current position
        if(matrix[i][j] == '1'){
            int ans = 1 + min(right, min(diagonal, down));
            maxi = max(maxi, ans);
            return ans;
        }
        else{
            // agar matrix[i][j] == 0 hai to ans hi zero hoga
            return 0;
        }
    }
    int maximalSquare(vector<vector<char>>& matrix) {
        int i = 0;
        int j = 0;
        int row = matrix.size();
        int col = matrix[0].size();
        int maxi = 0;

        int ans = solve(matrix, i, j, row, col, maxi);

        int area =  maxi*maxi;
        return area;
    }
};
```

DRY RUN = ?

T.C. = ?

S.C. = ?