1277. Count Square Submatrices with All Ones

```
Input: matrix =
[
  [0,1,1,1],
  [1,1,1,1],
  [0,1,1,1]
]
Output: 15
```

ans = 0

ans = 1̶ 2̶ 3̶ 4̶ 5̶ 6̶ 7̶ 8̶ 10 12 (15)

	0	1	2	3
0	0	1	1	1
1	1	1	2	2
2	0	1	2	3

invalid → 1

$$\min(\ A[i-1][j],\ A[i][j-1],\ A[i-1][j-1]\ ) + 1$$

$$(\quad 2 \qquad 2 \qquad 2\quad)$$

$$2 + 1$$

3×3
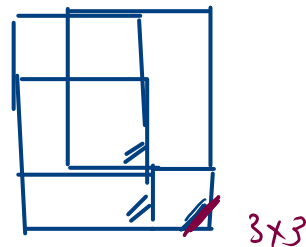
```java
class Solution {
    public int countSquares(int[][] matrix) {
        int m = matrix.length;
        int n = matrix[0].length;

        int ans = 0;

        for(int i = 0; i < m; i++){
            for(int j = 0; j < n; j++){
                if(matrix[i][j] == 1 && i-1 >= 0 && j-1 >= 0){
                    int min = Math.min(matrix[i-1][j-1], Math.min(matrix[i-1][j], matrix[i][j-1]));
                    matrix[i][j] = min + 1;
                }
                ans += matrix[i][j];
            }
        }
    }
```

Input: matrix =

[

  [0,1,1,1],

  [1,1,1,1],

  [0,1,1,1]

]

Output: 15

ms = 0 1 2 3 4 5 7
      8 9 10
      12 (15)

n + 0 = 2

i , j

1, 2

1-1 ≥ 0 ✓
2-1 ≥ 0 ✓

2+1

18. 4Sum

Medium 👍 9925 👎 1182 ♡ Add to List ⎙ Share

Given an array `nums` of n integers, return *an array of all the **unique***
quadruplets `[nums[a], nums[b], nums[c], nums[d]]` such that:

- `0 <= a, b, c, d < n`
- `a`, `b`, `c`, and `d` are **distinct**.
- `nums[a] + nums[b] + nums[c] + nums[d] == target`
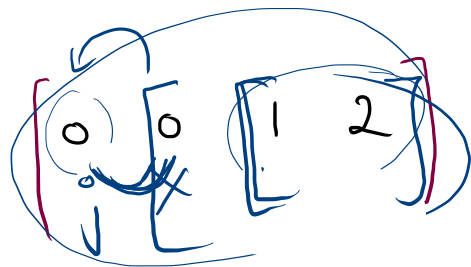
You may return the answer in **any order**.

Example 1:

```
Input: nums = [1,0,-1,0,-2,2], target = 0
Output: [[-2,-1,1,2],[-2,0,0,2],[-1,0,0,1]]
```

< ⟨ -2 ,1, 2, -1⟩

⟩

-2   -1   0   0   1   2

i

tar = 0

nt = 2