1504. Count Submatrices With All Ones

Medium | Topics | Companies | Hint

Given an `m x n` binary matrix `mat`, *return the number of* ***submatrices*** *that have all ones.*

Example 1:

1	0	1
1	1	0
1	1	0

```
Input: mat = [[1,0,1],[1,1,0],[1,1,0]]
Output: 13
Explanation:
There are 6 rectangles of side 1x1.
There are 2 rectangles of side 1x2.
There are 3 rectangles of side 2x1.
There is 1 rectangle of side 2x2.
There is 1 rectangle of side 3x1.
Total number of rectangles = 6 + 2 + 3 + 1 + 1 = 13.
```

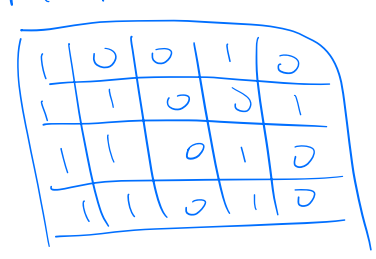
Method 1 : enumeration

- straightforward.
- count how many submatrix with this current position mat[i][j] as bottom-right corner could have

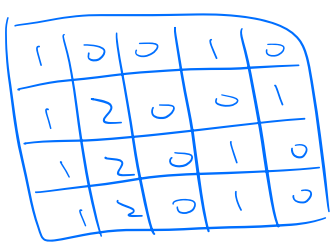
$$row[i][j] = \begin{cases} 0 & \text{if } mat[i][j] == 0 \\ row[i][j-1] + 1 & \text{if } mat[i][j] == 1 \end{cases}$$

by # of consecutive 1s extending to the left from position (i, j) in the matrix

mat



row



1×1 : 11 2×1 : 3
1×2 : 6 2×2 : 2
1×3 : 3 2×3 : 1

Once row[][] is ready, use (i, j) as bottom-right corner, enumerate height of the sub-rectangles and check how many valid.

for (i-1)th row, min(row[i][j], row[i-1][j]) sub-rectangles there.

```python
class Solution:
    def numSubmat(self, mat: List[List[int]]) -> int:
        m, n = len(mat), len(mat[0])
        res = 0
        row = [[0] * n for _ in range(m)]
        for i in range(m):
            for j in range(n):
                if j == 0:
                    row[i][j] = mat[i][j]
                else:
                    row[i][j] = 0 if mat[i][j] == 0 else row[i][j - 1] + 1
                cur = row[i][j]
                for k in range(i, -1, -1):
                    cur = min(cur, row[k][j])
                    if cur == 0:
                        break
                    res += cur
        return res
```

先 horizontal 扫

再 vertical 扫

在每个 (i, j) 时都计一次

row

1	0	0	1	0
1	2	0	0	1
1	2	0	1	0
1	2	0	1	0

res =

1		1	1	1	
1+1	2	1	2	2	1
1+1+1	2+2	1	3	4	1
1+1+1+1	2+1+2	1+1	4	6	2

27 ✓

time: $O(m^2 \cdot n)$

space: $O(m \cdot n)$