

# 最大正方形

题目：

221. 最大正方形

难度 中等

535

收藏

分享

切换为英文

关注

反馈

在一个由 0 和 1 组成的二维矩阵内，找到只包含 1 的最大正方形，并返回其面积。

示例：

输入：

```
1 0 1 0 0
1 0 1 1 1
1 1 1 1 1
1 0 0 1 0
```

输出：4

思路：

最大正方形，正方形是长 = 宽，

子问题：

如果  $matrix[i][j] == 1$ ，那么它的上方，左方，左上方都是 1；

现用  $dp(i, j)$  表示以  $(i, j)$  为右下角，且只包含 1 的正方形最大边长值，

如果该位置值为 0，则  $dp(i, j) = 0$ ，因为当前位置不可能在由 1 组成的正方形

如果该位置值为 1，则确定其上方，左方，左上方三个相邻位置的  $dp$  值，则由当前位置的元素值由这三个相邻位置的元素中最小值 + 1，

```
1 dp[i][j] = min(dp[i-1][j], dp[i][j-1], dp[i-1][j-1]) + 1
```

原始矩阵

	0	1	2	3	4
0	0	1	1	1	0
1	1	1	1	1	0
2	0	1	1	1	1
3	0	1	1	1	1
4	0	0	1	1	1

3 × 3 表示  $dp[2][3]$

2 × 2 表示  $dp[3][4]$

1 × 1 表示  $dp[4][2]$

dp

	0	1	2	3	4
0	0	1	1	1	0
1	1	1	2	2	0
2	0	1	2	3	1
3	0	1	2	3	2
4	0	0	1	2	3

$dp(2, 3) = \min(dp(1, 3), dp(1, 2), dp(2, 2)) + 1 = 3$

$dp(3, 4) = \min(dp(2, 4), dp(2, 3), dp(3, 3)) + 1 = 2$

$dp(4, 2) = \min(dp(3, 2), dp(3, 1), dp(4, 1)) + 1 = 1$

```
1 def maximalSquare(self, matrix: List[List[str]]) -> int:
```

```

2         if len(matrix) == 0 or len(matrix[0]) == 0:
3             return 0
4
5         maxSide = 0
6         rows, cols = len(matrix), len(matrix[0])
7         dp = [[0] * cols for _ in range(rows)]
8         for i in range(rows):
9             for j in range(cols):
10                if matrix[i][j] == 1:
11                    if i == 0 and j == 0 :
12                        dp[i][j] = 1
13                    else:
14                        dp[i][j] = min(dp[i-1][j], dp[i][j-1], dp[i-1]
[j -1]) +1
15                maxSide = max(maxSide, dp[i][j])
16        return maxSide * maxSide

```