1183. Maximum Number of Ones

My Submissions (/contest/biweekly-contest-8/problems/maximum-number-of-ones/submissions/)

Back to Contest (/contest/biweekly-contest-8/)

Consider a matrix `M` with dimensions `width * height`, such that every cell has value `0` or `1`, and any **square** sub-matrix of `M` of size `sideLength * sideLength` has at most `maxOnes` ones.

Return the maximum possible number of ones that the matrix `M` can have.

User Accepted:	61
User Tried:	188
Total Accepted:	67
Total Submissions:	379
Difficulty:	Hard

Example 1:

```
Input: width = 3, height = 3, sideLength = 2, maxOnes = 1
Output: 4
Explanation:
In a 3*3 matrix, no 2*2 sub-matrix can have more than 1 one.
The best solution that has 4 ones is:
[1,0,1]
[0,0,0]
[1,0,1]
```

Example 2:

```
Input: width = 3, height = 3, sideLength = 2, maxOnes = 2
Output: 6
Explanation:
[1,0,1]
[1,0,1]
[1,0,1]
```

Constraints:

- `1 <= width, height <= 100`
- `1 <= sideLength <= width, height`
- `0 <= maxOnes <= sideLength * sideLength`

Discuss (https://leetcode.com/problems/maximum-number-of-ones/discuss)

Python3

Load successfully.

```
class Solution:
    def maximumNumberOfOnes(self, C, R, K, maxOnes):
        # every K*K square has at most maxOnes ones
        count = [0] * (K*K)
```

```
        for r in range(R):
            for c in range(C):
                # calculate index to transform from 2D to 1D
                index = (r%K) * K + c%K
                count[index] += 1
        print(count)
        count.sort()
        ans = 0
        for _ in range(maxOnes):
            ans += count.pop()
        return ans
```

☐ **Custom Testcase**

Run Code Submit