Problem 1183: Maximum Number of Ones

Problem Information

Difficulty: Hard
Acceptance Rate: 71.12%
Paid Only: Yes
Tags: Math, Greedy, Sorting, Heap (Priority Queue)

Problem Description

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.

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`

Code Snippets

C++:

```
class Solution {
public:
int maximumNumberOfOnes(int width, int height, int sideLength, int maxOnes) {

}
};
```

Java:

```
class Solution {
public int maximumNumberOfOnes(int width, int height, int sideLength, int
maxOnes) {

}
}
```

Python3:

```
class Solution:
def maximumNumberOfOnes(self, width: int, height: int, sideLength: int,
maxOnes: int) -> int:
```