

Oj.1002 子矩阵最大值

2019年1月21日 星期一 下午8:30

1002. 二哥种花生

Description

二哥在自己的后花园里种了一些花生，也快到了收获的时候了。这片花生地是一个长度为L、宽度为W的矩形，每个单位面积上花生产量都是独立的。他想知道，对于某个指定的区域大小，在这么大的矩形区域内，花生的产量最大会是多少。

Input Format

第1行有2个整数，长度L和宽度W。

第2行至第L+1行，每行有W个整数，分别表示对应的单位面积上的花生产量A（ $0 \leq A < 10$ ）。

第L+2行有2个整数，分别是指定的区域大小的长度a和宽度b。

Output Format

输出一个整数m，表示在指定大小的区域内，花生最大产量为m。

Sample Input

```
4 5
1 2 3 4 5
6 7 8 0 0
0 9 2 2 3
3 0 0 0 1
3 3
```

Sample Output

```
38
```

样例解释

左上角： $38 = (1+2+3) + (6+7+8) + (0+9+2)$

数据范围

对于30%的数据： $1 \leq L, W \leq 100$;

对于100%的数据： $1 \leq L, W \leq 1000$ 。

全部区域大小满足： $1 \leq a \leq L, 1 \leq b \leq W$ 。

最初想法

框一个个遍历，一个个求和

Map $n * n$
Kernel $k * k$
 $O(n * n * k * k)$

超时

中间有很多运算重复了

正解

初始化矩阵sum

Sum[i][j]表示[1, 1] 到[i, j]的矩阵的和

$sum[i][j] = map[i][j] + sum[i - 1][j] + sum[i][j - 1] - sum[i - 1][j - 1];$

计算框的值时

$temp = sum[i][j] - sum[i - kernelRow][j] - sum[i][j - kernelCol] + sum[i - kernelRow][j - kernelCol];$

然后找里面的最大值

$O(n * n)$