题意就是在一个矩阵中求出实心矩形的个数。

我们先想想最暴力的做法：记黑点为1，红点为0，那么一个矩形为实心矩形的充要条件就是里面没有1。我们可以用一个二维前缀和在常数时间内得出任一矩形内的元素和。

如果枚举矩形的左上角与右上角，这个复杂度是，显然不够。

考虑我们正在枚举矩形的右下角，我们真的需要枚举所有横纵坐标小于右下角的点？

如果确定了矩形的右下角，那对于每一行而言，只有从右往左连续的一段红色方块才有可能对答案有贡献。

于是我们维护一个数组d,d[x][y]表示第x行y列的这个点，前面有多少个连续的红色方块。

如果我们确定了右下角在x2行y2列，而左上角在x1行，那么这一行可以作为左上角的点的数量为min{d[i][y]},x1<=i<=x2。我们枚举右下角，然后往上枚举左上角所在的行，即可做到复杂度，这个代码在题目包里也有。

但是这个复杂度显然还不够。。。

我们在想到上述算法之后，应该会发现，固定右上角之后，从x2到第一行，对答案的贡献依次是一个递减的阶梯状的值，这个值是可以用单调栈维护的。。。

于是我们压到了，这便是标程的做法。

PS：k出的很小，不知道有没有人想出了乱搞算法呢。。。。