**F - Search gold**

**Time Limit: 3000/1000MS (Java/Others)     Memory Limit: 65535/65535KB (Java/Others)**

Submit Status

Dreams of finding lost treasure almost came true recently. A new machine called 'The Revealer' has been invented and it has been used to detect gold which has been buried in the ground. The machine was used in a cave near the seashore where -- it is said -- pirates used to hide gold. The pirates would often bury gold in the cave and then fail to collect it. Armed with the new machine, a search party went into the cave hoping to find buried treasure. The leader of the party was examining the soil near the entrance to the cave when the machine showed that there was gold under the ground. Very excited, the party dug a hole two feel deep. They finally found a small gold coin which was almost worthless. The party then searched the whole cave thoroughly but did not find anything except an empty tin trunk. In spite of this, many people are confident that 'The Revealer' may reveal something of value fairly soon.

So,now you are in the point(1,1)(1,1) and initially you have 0 gold.In the nn\*mm grid there are some traps and you will lose gold.If your gold is not enough you will be die.And there are some treasure and you will get gold.If you are in the point(x,y),you can only walk to point (x+1,y),(x,y+1),(x+1,y+2)(x+1,y),(x,y+1),(x+1,y+2)and(x+2,y+1)(x+2,y+1).Of course you can not walk out of the grid.Tell me how many gold you can get most in the trip.

It`s guarantee that(1,1)(1,1)is not a trap;

**Input**

first come 22 integers, n,mn,m(1≤n≤10001≤n≤1000,1≤m≤10001≤m≤1000)

Then follows nn lines with mm numbers aijaij

(−100<=aij<=100)(−100<=aij<=100)

the number in the grid means the gold you will get or lose.

**Output**

print how many gold you can get most.

**Sample input and output**

| **Sample Input** | **Sample Output** |
| --- | --- |
| 3 3  1 1 1  1 -5 1  1 1 1 | 5 |
| 3 3  1 -100 -100  -100 -100 -100  -100 -100 -100 | 1 |

题意：

在n\*m的坐标系中，每个整数坐标都有一个权值，从当前格子(i,j)只能前往(i+1,j), (i,j+1), (i+1,j+2), (i+2,j+1),每条路径的权值为整个路径上所有点的权值和，且必须满足：

1. 路径从(1,1)出发；
2. 从(1,1)到路径上任意一点路径的权值和大于0；

问所有这样的路径中最大的权值是多少。

题解：

首先，这道题很明显可以利用搜索来解决。考虑到此题点数较多，如果dfs很有可能栈溢出，所以最好使用bfs来解决问题。不过，bfs又会出现新的问题：状态过多，导致队列过长，空间复杂度较高。

此时观察到此题的一个特点：只能向右或向下走，也就是说，我们如果从(1,1)开始一行一行扫描，则路径终点为当前点的权值必定只和已经扫描过的点有关，因为要想到当前点，只能从当前点左上方过来。因此可以利用DP，从(1,1)开始扫描每个点，如果当前点的最优解大于0则依次穷举四个方向，得到到达后续节点的最优路径，并利用当前点的最优解更新ans。时间复杂度O(nm)。

此外还有一点要注意，那就是必须从(1,1)出发。另定义一个标记数组记录每个点是否可以到达即可。