

题目描述：

从一个 $N \times M$ ($N \leq M$) 的矩阵中选出 N 个数，任意两个数字不能在同一行或同一列，求选出来的 N 个数中第 K 大的数字的最小值是多少。

输入描述：

输入矩阵要求： $1 \leq K \leq N \leq M \leq 150$

输入格式：

$N \ M \ K$

$N \times M$ 矩阵

输出描述：

$N \times M$ 的矩阵中可以选出 $M!/N!$ 种组合数组，每个组合数组中第 K 大的数中的最小值。无需考虑重复数字，直接取字典排序结果即可。

补充说明：

注意：结果是第 K 大的数字的最小值

示例 1

输入：

3 4 2
1 5 6 6
8 3 4 3
6 8 6 3

输出：

3

说明：

$N \times M$ 的矩阵中可以选出 $M!/N!$ 种组合数组，每个组合数组中第 K 大的数中的最小值；上述输入中选出的数组组合为 1,3,6; 1,3,3; 1,4,8; 1,4,3;.....

上述输入样例中选出的组合数组有 24 种，最小数组为 1,3,3，则 2 大的最小值为 3

def check(p, mp, n, m, K):

```

mp = [[mp[i][j] <= p for j in range(m + 1)] for i in range(n + 1)]
ans = 0
match = [0] * (m + 1)
for i in range(1, n + 1):
    chw = [True] * (m + 1)
    if find_muniu(i, mp, chw, match):
        ans += 1
if ans >= n - K + 1:
    return True
return False

```

```

def solve(mx):
    l, r = 1, mx
    ans = 0

    while l <= r:
        mid = (l + r) // 2
        if check(mid, my_map, n, m, K):
            ans = mid
            r = mid - 1
        else:
            l = mid + 1
    return ans

```

```

def find_muniu(x, mp, chw, match):
    for j in range(1, m + 1):
        if mp[x][j] and chw[j]:
            chw[j] = False
            if match[j] == 0 or find_muniu(match[j], mp, chw, match):
                match[j] = x
                return True
    return False

```

```

mx = 0
n, m, K = map(int, input().split())
my_map = [[0] * (m + 1) for _ in range(n + 1)]

for i in range(1, n + 1):
    row = list(map(int, input().split()))
    for j in range(1, m + 1):
        my_map[i][j] = row[j - 1]

```

```
mx = max(mx, my_map[i][j])
```

```
print(solve(mx))
```