

- [题单简介](#)
  - [完整的题单戳这里](#)
  - [题目描述](#)
  - [输入格式](#)
  - [输出格式](#)
  - [样例 #1](#)
    - [样例输入 #1](#)
    - [样例输出 #1](#)
  - [提示](#)
    - [样例解释](#)
    - [数据范围](#)

## 题单简介

---

2020.7.1 UPD:

这份题单中大部分是模版题和经典题

主要可以帮助你了解oi中有哪些知识点，构建起知识框架

如果希望熟练的掌握某个算法还请另找相关题目多加练习

另：如果你仅参加NOIP而并不打算参加省选或省选以上级别的比赛，这份题单中许多算法并不是你必须掌握的

建议了解清楚后学习

希望这份题单能成为你oi生涯中有效的助力(

有什么修改建议可以私信我(

2020.8.29Upd:感谢[jwj](#)提供了新的排版

---

- 因为没有算法大纲而迷茫，不知道学什么？
- 题海茫茫，找不到典型/合适的例题练手？

这份 科学、全面 的能力进阶综合题单一定能帮到你！

这份题单包含了竞赛中全部的常用算法和技巧，配以典型的例题用以练手

新手可以把它作为学习方向的引导，神犇可以把它作为查漏补缺和复习的大纲

为了方便按知识难度刷题的用户，这里给出一些建议：

- 对于初学者，建议先完成 **Part 1,2** 两部分内容，为接下来的学习打好基础。
- 对于要参加 **CSP-S/NOIP提高组** 的选手，建议在前面的基础上优先完成 **Part 3.1-3.4, 4.1-4.4, 6.1-6.5, 7.1-7.8, 8.1-8.7** 的内容（具体内容见下），在此基础上继续完成其他内容。
- 每个专题下的题目先给出模板，剩下的题目均按照难度递增顺序排序，部分难度较高的综合性题目建议达到一定能力后再尝试解决。

## 完整的题单戳[这里](#)

---

### Part 1 入门阶段

- Part 1.1 从零开始
- Part 1.2 数组基础
- Part 1.3 字符串基础
- Part 1.4 函数，递归及递推

### Part 2 基础算法

- Part 2.1 模拟
- Part 2.2 排序算法
- Part 2.3 二分答案
- Part 2.4 分治
- Part 2.5 贪心
- Part 2.6 构造
- Part 2.7 高精度
- Part 2.8 前缀和 & 差分

### Part 3 搜索

- Part 3.1 深度优先搜索
- Part 3.2 广度优先搜索
- Part 3.3 记忆化搜索
- Part 3.4 搜索的剪枝
- Part 3.5 双向搜索
- Part 3.6 A\*
- Part 3.7 IDA\*
- Part 3.8 DLX

### Part 4 动态规划

- [Part 4.1-4.4 动态规划](#)
  - Part 4.1 线性动态规划
  - Part 4.2 背包动态规划
  - Part 4.3 区间动态规划
  - Part 4.4 树形动态规划
- [Part 4.5-4.12 动态规划](#)
  - Part 4.5 状态压缩动态规划
  - Part 4.6 倍增优化动态规划
  - Part 4.7 数据结构优化动态规划
  - Part 4.8 单调队列优化动态规划
  - Part 4.9 斜率优化动态规划
  - Part 4.10 决策单调性优化动态规划
  - Part 4.11 数位统计类动态规划
  - Part 4.12 轮廓线动态规划

## Part 5 字符串

- [Part 5.1-5.6 字符串](#)
  - Part 5.1 字符串哈希
  - Part 5.2 KMP
  - Part 5.3 Manacher
  - Part 5.4 Trie树
  - Part 5.5 AC自动机
  - Part 5.6 回文自动机
- [Part 5.7-5.8 字符串](#)
  - Part 5.7 后缀数组
  - Part 5.8 后缀自动机

## Part 6 数学

- [Part 6.1-6.4 数学](#)
  - Part 6.1 位运算
  - Part 6.2 整除相关
    - Part 6.2.1 素数
    - Part 6.2.2 最大公约数
    - Part 6.2.3 欧拉函数
  - Part 6.3 同余方程
    - Part 6.3.1 线性同余方程&乘法逆元
    - Part 6.3.2 中国剩余定理
    - Part 6.3.3 高次同余方程

- Part 6.4 博弈论
- [Part 6.5-6.6 数学](#)
  - Part 6.5 概率与期望
  - Part 6.6 组合数学
    - Part 6.6.1 排列组合
    - Part 6.6.2 卡特兰数&斯特林数
    - Part 6.6.3 容斥原理
- [Part 6.7-6.8 数学](#)
  - Part 6.7 线性代数
    - Part 6.7.1 矩阵
    - Part 6.7.2 高斯消元
    - Part 6.7.3 线性基
  - Part 6.8 多项式
- [Part 6.9-6.13 数学](#)
  - Part 6.9 莫比乌斯反演
  - Part 6.10 筛法
  - Part 6.11 线性规划
  - Part 6.12 数值方法
    - Part 6.12.1 三分法
    - Part 6.12.2 自适应辛普森法
  - Part 6.13 置换群

## Part 7 数据结构

- [Part 7.1-7.7 数据结构](#)
  - Part 7.1 链表
  - Part 7.2 栈
  - Part 7.3 队列
  - Part 7.4 并查集
  - Part 7.5 二叉堆
  - Part 7.6 ST表
  - Part 7.7 树状数组
- [Part 7.8-7.12 数据结构](#)
  - Part 7.8 线段树
  - Part 7.9 分块
  - Part 7.10 可并堆
  - Part 7.11 主席树
  - Part 7.12 平衡树
- [Part 7.13-7.18 数据结构](#)

- Part 7.13 树链剖分
- Part 7.14 树套树
- Part 7.15 动态树
- Part 7.16 可持久化数据结构
- Part 7.17 K-D Tree
- Part 7.18 珂朵莉树

## Part 8 图论

- [Part 8.1-8.6 图论](#)
  - Part 8.1 图的存储与遍历
  - Part 8.2 最短路问题
  - Part 8.3 树上问题
    - Part 8.3.1 二叉树
    - Part 8.3.2 树的直径
    - Part 8.3.3 最近公共祖先
  - Part 8.4 生成树
  - Part 8.5 拓扑排序
  - Part 8.6 差分约束
- [Part 8.7-8.9.2 图论](#)
  - Part 8.7 图的连通性相关
  - Part 8.8 二分图
  - Part 8.9 网络流
    - Part 8.9.1 最大流
    - Part 8.9.2 最小割
- [Part 8.9.3-8.13 图论](#)
  - Part 8.9.3 费用流
  - Part 8.9.4 上下界网络流
  - Part 8.10 2-SAT
  - Part 8.11 点分治
  - Part 8.12 虚树
  - Part 8.13 矩阵树定理

## Part 9 计算几何

- Part 9.1 凸包
- Part 9.2 旋转卡壳
- Part 9.3 半平面交

## Part 10 杂项

- Part 10.1 模拟退火
- Part 10.2 0/1 分数规划
- Part 10.3 离线算法
  - Part 10.3.1 CDQ 分治
  - Part 10.3.2 整体二分
  - Part 10.3.3 莫队
- Part 10.4 奇怪的题目
- Part 10.5 非传统题
  - Part 10.5.1 提交答案

## 题目描述

跳蚤国王和蚰蚰国王在玩一个游戏。

他们在一个  $n$  行  $m$  列的网格上排兵布阵。其中的  $c$  个格子中 ( $0 \leq c \leq n \cdot m$ )，每个格子有一只蚰蚰，其余的格子中，每个格子有一只跳蚤。

我们称占据的格子有公共边的两只跳蚤是相邻的。

我们称两只跳蚤是连通的，当且仅当这两只跳蚤相邻，或存在另一只跳蚤与这两只跳蚤都连通。

现在，蚰蚰国王希望，将某些（零个，一个或多个）跳蚤替换成蚰蚰，使得在此之后存在至少两只跳蚤不连通。

例如：图 1 描述了一个  $n = 4, m = 4, c = 2$  的情况。

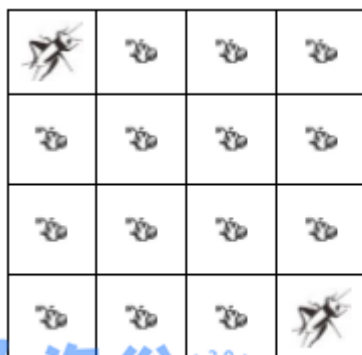


图 1

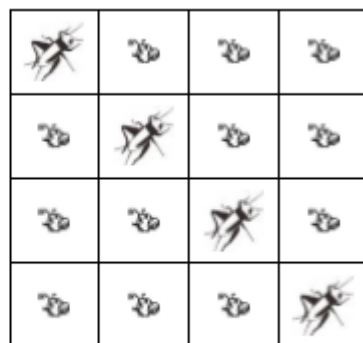


图 2

这种情况下蚰蚰国王可以通过将第二行第二列，和第三行第三列的两只跳蚤替换为蚰蚰，从而达成他的希望，如右图所示。并且，不存在更优的方案，但是可能存在其他替换两只跳蚤的方案。

你需要首先判断蚰蚰国王的希望能否被达成。如果能够达成，你还需要最小化被替换的跳蚤的个数。

## 输入格式

---

每个输入文件包含多组数据。

输入文件的第一行只有一个整数  $T$ ，表示数据的组数。

接下来依次输入  $T$  组数据，每组数据的第一行包含三个整数  $n, m, c$ 。

接下来  $c$  行，每行包含两个整数  $x, y$  表示第  $x$  行，第  $y$  列的格子被一个蚰蚰占据。每一组数据当中，同一个蚰蚰不会被多次描述。

## 输出格式

---

对于每一组数据依次输出一行答案。

如果这组数据中，蚰蚰国王的希望不能被达成，输出  $-1$ 。否则，输出被替换的跳蚤的个数的最小值。

## 样例 #1

---

### 样例输入 #1

```
4
4 4 2
1 1
4 4
2 3 1
1 2
2 2 2
1 1
2 2
1 1 0
```

### 样例输出 #1

2  
1  
0  
-1

# 提示

## 样例解释

第一组数据就是问题描述中的例子。

对于第二组数据，可以将第二行第二列的一只跳蚤替换为蚰蚰，从而使得存在两只跳蚤不连通，并且不存在更优的方案。

对于第三组数据，最初已经存在两只跳蚤不连通，故不需要再进行替换。

对于第四组数据，由于最多只有一只跳蚤，所以无论如何替换都不能存在两只跳蚤不连通。

## 数据范围

对于全部的测试点，保证  $1 \leq T \leq 20$ 。我们记  $\sum c$  为某个测试点中，其  $T$  组输入数据的所有  $c$  的总和。对于所有的测试点， $\sum c \leq 10^5$ 。

对于全部的数据，满足  $1 \leq n, m \leq 10^9$ ， $0 \leq c \leq n \times m$ ， $1 \leq x \leq n, 1 \leq y \leq m$ 。

每个测试点的详细数据范围见下表。表中的  $n, m, c$  均是对单个输入数据（而非测试点）而言的，也就是说同一个测试点下的  $T$  组数据均满足限制条件；而  $\sum c$  是对单个测试点而言的。为了方便阅读，“测试点”一列被放到了表格的中间而不是左边。

$n, m$	测试点	$c$
$n * m \leq 4$	1	$c \leq n * m$
$n * m \leq 8$	2	$c \leq n * m$
$n * m \leq 15$	3	$c \leq n * m$
$n * m \leq 30$	4	$c \leq n * m$
$n * m \leq 100$	5	$c \leq n * m$



$n, m$	测试点	$c$
$n * m \leq 300$	6	$c \leq n * m$
$n * m \leq 10^3$	7	$c \leq n * m$
$n * m \leq 2 \times 10^4$	8	$c \leq 5$
$n * m \leq 2 \times 10^4$	9	$c \leq 15$
$n * m \leq 2 \times 10^4$	10	$c \leq 30$
$n, m \leq 2 \times 10^4, n * m \leq 2 \times 10^4$	11	$\sum c \leq 2 \times 10^4$
$n, m \leq 2 \times 10^4, n * m \leq 10^5$	12	$\sum c \leq 2 \times 10^4$
$n, m \leq 2 \times 10^4, n * m \leq 3 \times 10^5$	13	$\sum c \leq 2 \times 10^4$
$n, m \leq 2 \times 10^4, n * m \leq 10^6$	14	$\sum c \leq 2 \times 10^4$
$n, m \leq 2 \times 10^4, n * m \leq 10^9$	15	$\sum c \leq 2 \times 10^4$
$n, m \leq 10^5$	16	$\sum c \leq 10^5$
$n, m \leq 10^9$	17	$c = 0$
$n, m \leq 10^9$	18	$c \leq 1$
$n, m \leq 10^9$	19	$c \leq 2$
$n, m \leq 10^9$	20	$c \leq 3$
$n, m \leq 10^9$	21	$c \leq 10$
$n, m \leq 10^9$	22	$c \leq 30$
$n, m \leq 10^9$	23	$c \leq 300$
$n, m \leq 10^9$	24	$\sum c \leq 2 \times 10^4$
$n, m \leq 10^9$	25	$\sum c \leq 10^5$