2023 广东省大学生程序设计竞赛

正式赛

2023 年 5 月 14 日

试题列表

A	算法竞赛
B	基站建设
C	市场交易
D	新居规划
E	新怀质问
F	格子旅行
G	交换操作
H	流画溢彩
I	路径规划
J	X 等于 Y
K	独立钻石
L	经典问题
M	计算几何

本试题册共 13 题，21 页。
如果您的试题册缺少页面，请立即通知志愿者。

由 SUA 程序设计竞赛命题组命题。
https://sua.ac/

承办方

命题方

竞赛过程中访问非竞赛网页是违反竞赛规则的行为。
如果您有兴趣（我们很荣幸），
请在竞赛后扫描二维码。

Problem A. 算法竞赛

广东省是全国较早一批将程序设计竞赛引入省内大学生竞赛体系的省份之一。2003 年，中山大学承办了第一届广东省大学生程序设计竞赛。此后，华南农业大学、华南理工大学、华南师范大学等省内高校也先后承办了此赛事，除 2020 年因疫情停办外，每年一届。2023 年，深圳技术大学将承办第二十届广东省大学生程序设计竞赛，让我们期待选手们出色的表现！

图：深圳技术大学的美丽校园。

在另一个世界中，某程序设计竞赛自 y_1 年起开始举办。除了 $s_1, s_2, \cdots, s_n$ 这 n 年由于特殊原因无法举办之外，其他年份每年举办一次。

求 y_2 年是该竞赛的第几次举办。

Input

有多组测试数据。第一行输入一个整数 T（$1 \le T \le 20$）表示测试数据组数。对于每组测试数据：

第一行输入一个整数 y_1（$1970 \le y_1 \le 9999$），表示该竞赛第一次举办的年份。

第二行首先输入一个整数 n（$0 \le n \le 100$）表示该竞赛停办的年份数，之后输入 n 个整数 $s_1, s_2, \cdots, s_n$（$y_1 < s_i \le 9999$）表示该竞赛的停办年份。停办年份按递增顺序给出，且没有重复的年份。

第三行输入一个整数 y_2（$y_1 \le y_2 \le 9999$）。保证 y_2 不是停办年份之一。

Output

每组数据输出一行一个整数，表示 y_2 年是该竞赛的第几次举办。

Example

standard input	standard output
4 2003 1 2020 2023 2003 1 2020 2003 2345 0 3456 3000 4 3001 3003 3004 3008 3007	20 1 1112 5

Note

对于第一组样例数据，如题目描述中所述，答案为 20。

对于第二组样例数据，由于 2003 就是该竞赛第 1 次举办的年份，因此答案为 1。

对于第三组样例数据，由于竞赛从未停办，因此答案为 $3456 - 2345 + 1 = 1112$。

对于第四组样例数据，该竞赛前 5 次举办的年份为 3000，3002，3005，3006 与 3007。因此答案为 5。

Problem B. 基站建设

中国移动通信集团广东有限公司深圳分公司（以下简称“深圳移动”）于 1999 年正式注册。四年后，广东省大学生程序设计竞赛第一次举办。深圳移动与广东省大学生程序设计竞赛一起见证了广东省计算机行业的兴旺与发展。

图：中国移动 5G 基站。

在建设通信线路的过程中，信号基站的选址是一个非常关键的问题。某城市从西到东的距离为 n 千米，工程师们已经考察了在从西往东 $1, 2, \cdots, n$ 千米的位置建设基站的成本，分别是 $a_1, a_2, \cdots, a_n$。

为了保证居民的通信质量，基站的选址还需要满足 m 条需求。第 i 条需求可以用一对整数 l_i 和 r_i 表示（$1 \le l_i \le r_i \le n$），代表从西往东 l_i 千米到 r_i 千米的位置之间（含两端）至少需要建设 1 座基站。

作为总工程师，您需要决定基站的数量与位置，并计算满足所有需求的最小总成本。

Input

有多组测试数据。第一行输入一个整数 T 表示测试数据组数，对于每组测试数据:

第一行输入一个整数 n（$1 \le n \le 5 \times 10^5$）表示城市从西到东的距离。

第二行输入 n 个整数 $a_1, a_2, \cdots, a_n$（$1 \le a_i \le 10^9$），其中 a_i 表示在从西往东 i 千米的位置建设基站的成本。

第三行输入一个整数 m（$1 \le m \le 5 \times 10^5$）表示需求的数量。

对于接下来 m 行，第 i 行输入两个整数 l_i 和 r_i（$1 \le l_i \le r_i \le n$）表示从西往东 l_i 千米到 r_i 千米的位置之间（含两端）至少需要建设 1 座基站。

保证所有数据 n 之和与 m 之和均不超过 5×10^5。

Output

每组数据输出一行一个整数，表示满足所有需求的最小总成本。

Example

standard input	standard output
2 5 3 2 4 1 100 3 1 3 2 4 5 5 5 7 3 4 2 2 3 1 4 2 3 4 5	102 5

Note

对于第一组样例数据，最优方案是在从西往东 2 千米和 5 千米的位置建设基站。总成本为 $2 + 100 = 102$。

对于第二组样例数据，最优方案是在从西往东 2 千米和 4 千米的位置建设基站。总成本为 $3 + 2 = 5$。

Problem C. 市场交易

二十年前，广州的北京路步行街北段出土了自唐代直到民国时期的十一层路面，南段则发掘出宋代至明清时期共五层的拱北楼建筑基址，佐证了北京路自宋代以来作为商业步行街的悠久历史；同时第一届广东省大学生程序设计竞赛也在位处广州的中山大学举办。二十年后的今天，北京路步行街已成为广州最负盛名的景点和购物胜地之一，而广东省大学生程序设计竞赛也迎来了自己的二十岁生日。

图：广州的北京路步行街。

在步行街中，有 n 间商店买卖同一种商品，第 i 间商店一件商品的收购价和出售价均为 a_i 元。为了防止过度交易，步行街有一个规定：您在第 i 间商店最多进行 b_i 次交易（一次买或一次卖均计为一次交易），且每次只能交易一件商品。

您准备通过在步行街中买卖这种商品来赚钱。假如初始时有无限的金钱（也就是说，不会因为钱不够而买不了一件商品），您最多能在步行街中赚到多少总利润？具体来说，“利润”指的是卖出商品获得的金钱总额，减去购买商品花费的金钱总额。

Input

有多组测试数据。第一行输入一个整数 T 表示测试数据组数。对于每组测试数据：

第一行输入一个整数 n（$1 \le n \le 10^5$），表示商店的数量。

对于接下来 n 行，第 i 行输入两个整数 a_i 和 b_i（$1 \le a_i, b_i \le 10^6$），分别表示第 i 间商店的商品价格，以及该商店可以交易的最大次数。

保证所有数据 n 之和不超过 10^6。

Output

每组数据输出一行一个整数，表示在步行街中赚到的最大总利润。

Example

standard input	standard output
2 4 10 2 30 7 20 4 50 1 2 1 100 1 1000	100 0

Note

对于第一组样例数据，最优方案是在第 1 间商店买入 2 件商品，在第 3 间商店买入 4 件商品，在第 2 间商店卖出 5 件商品，在第 4 间商店卖出 1 件商品。总利润为 $30 \times 5 + 50 \times 1 - 10 \times 2 - 20 \times 4 = 100$。

对于第二组样例数据，由于所有商店的商品价格都相同，因此无法获得利润。

Problem D. 新居规划

随着广东的建设与发展，越来越多人选择来到广东开始新生活。在一片新建的小区，有 n 个人要搬进 m 栋排成一行的房子，房子的编号从 1 到 m（含两端）。房子 u 和 v 相邻，当且仅当 $|u-v|=1$。我们需要为每一个人安排一栋房子，要求所有人入住的房子互不相同。若两个人住进了一对相邻的房子，则这两个人互为邻居。

有的人喜欢自己有邻居，而有的人不喜欢。对于第 i 个人，如果他有至少一位邻居，则他的满意度为 a_i；否则如果他没有邻居，则他的满意度为 b_i。

您作为小区的规划者，需要最大化所有人的总满意度。

Input

有多组测试数据。第一行输入一个整数 T 表示测试数据组数。对于每组测试数据：

第一行输入两个整数 n 和 m（$1 \le n \le 5 \times 10^5$，$1 \le m \le 10^9$，$n \le m$），表示人数和房子数。

对于接下来的 n 行，第 i 行输入两个整数 a_i 和 b_i（$1 \le a_i, b_i \le 10^9$），表示第 i 个人在有邻居和没有邻居时的满意度。

保证所有数据 n 之和不超过 10^6。

Output

每组数据输出一行一个整数表示最大总满意度。

Example

standard input	standard output
3 4 5 1 100 100 1 100 1 100 1 2 2 1 10 1 10 2 3 100 50 1 1000	400 2 1050

Note

对于第一组样例数据，最优方案是让第 1 个人入住房子 1，第 2 到 4 个人入住房子 3 到 5。这样，第 1 个人没有邻居，而第 2 到 4 个人都有邻居。答案为 $100+100+100+100=400$。当然，也可以让第 2 到 4 个人入住房子 1 到 3，第 1 个人入住房子 5，也能得到 400 的总满意度。

对于第二组样例数据，由于只有 2 栋房子，因此第 1 和第 2 个人必须成为邻居。答案为 $1+1=2$。

对于第三组样例数据，最优方案是让第 1 个人入住房子 1，第 2 个人入住房子 3。这样，两个人都没有邻居。答案为 $50+1000=1050$。

Problem E. 新怀质问

给定 n 个字符串 $w_1, w_2, \cdots, w_n$，请选出恰好 k 个字符串，最小化字符串 v 的字典序，并输出这个最优的字符串 v。其中 v 满足以下条件：v 是被选出的字符串中，某两个编号不同的字符串的最长公共前缀。而且，v 是所有满足条件的字符串中，字典序最大的字符串。

更正式地，令 $\mathbb{S}$ 表示一个大小为 k 的集合，集合中的元素均为从 1 到 n 的整数（含两端），且没有重复的元素。令 $\mathrm{lcp}(w_i, w_j)$ 表示字符串 w_i 和 w_j 的最长公共前缀，您需要找到一个集合 $\mathbb{S}$ 以最小化下述字符串 v 的字典序，并输出这个最优的字符串 v。

$$v = \max_{i \in \mathbb{S}, j \in \mathbb{S}, i \neq j} \mathrm{lcp}(w_i, w_j)$$

上式中的 max 通过字典序比较两个字符串。

请回忆：

- 称字符串 p 是字符串 s 的前缀，若可以在 p 的末尾添加若干个字符（包括零个字符）将它变成 s。特别地，空字符串是任意字符串的前缀。
- 字符串 s 和 t 的最长公共前缀是一个最长的字符串 p，满足 p 既是 s 的前缀，又是 t 的前缀。例如，“abcde” 与“abcef” 的最长公共前缀为 “abc”，而 “abcde” 与 “bcdef” 的最长公共前缀为空字符串。
- 称字符串 s 的字典序小于字符串 t（$s \neq t$），若
  - s 是 t 的前缀，或
  - $s_{|p|+1} < t_{|p|+1}$，其中 p 为 s 和 t 的最长公共前缀，$|p|$ 为 p 的长度，s_i 表示字符串 s 的第 i 个字符，t_i 表示字符串 t 的第 i 个字符。

  特别地，空字符串是字典序最小的字符串。

Input

有多组测试数据。第一行输入一个整数 T 表示测试数据组数。对于每组测试数据：

第一行输入两个整数 n 和 k（$2 \le n \le 10^6$，$2 \le k \le n$），表示字符串的总数和需要选择的字符串的数量。

对于接下来 n 行，第 i 行输入一个由小写字母构成的字符串 w_i（$1 \le |w_i| \le 10^6$）。

保证所有数据中字符串长度之和不超过 10^6。

Output

每组数据输出一行一个字符串表示答案。特别地，若答案为空字符串，输出 `EMPTY`。

Example

standard input	standard output
2 5 3 gdcpc gdcpcpcp suasua suas sususua 3 3 a b c	gdcpc EMPTY

Problem F. 格子旅行

有 n 个格子排成一行，第 i 个格子的颜色为 c_i，上面放置着一个权值为 v_i 的球。

您将要在格子中进行若干次旅行。每次旅行时，您会得到旅行的起点 x 与一个颜色集合 $\mathbb{A} = \{a_1, a_2, \cdots, a_k\}$，且保证 $c_x \in \mathbb{A}$。旅行将从第 x 个格子上开始。在旅行期间，如果您在格子 i 处，那么您可以向格子 $(i-1)$ 或 $(i+1)$ 处移动，但不能移动到这 n 个格子之外。且在任意时刻，您所处的格子的颜色必须在集合 $\mathbb{A}$ 中。

当您位于格子 i 时，您可以选择将格子上的球取走，并获得 v_i 的权值。由于每个格子上只有一个球，因此一个格子上的球只能被取走一次。

您的任务是依次处理 q 次操作，每次操作形如以下三种操作之一：

- $1\ p\ x$：将 c_p 修改为 x。
- $2\ p\ x$：将 v_p 修改为 x。
- $3\ x\ k\ a_1\ a_2\ \ldots\ a_k$：给定旅行的起点 x 与一个颜色集合 $\mathbb{A} = \{a_1, a_2, \cdots, a_k\}$。假设如果进行这样的一次旅行，求出取走的球的权值之和最大是多少。注意，由于我们仅仅假设进行一次旅行，因此并不会真的取走任何球。即，所有询问之间是独立的。

Input

有多组测试数据。第一行输入一个整数 T 表示测试数据组数。对于每组测试数据：

第一行输入两个整数 n 和 q（$1 \le n \le 10^5$，$1 \le q \le 10^5$）表示格子的数量和操作的数量。

第二行输入 n 个整数 $c_1, c_2, \ldots, c_n$（$1 \le c_i \le n$），其中 c_i 表示第 i 个格子的初始颜色。

第三行输入 n 个整数 $v_1, v_2, \ldots, v_n$（$1 \le v_i \le 10^9$），其中 v_i 表示第 i 个格子里的球的初始权值。

对于接下来 q 行，第 i 行描述第 i 次操作，格式如下：

- $1\ p\ x$：保证 $1 \le p \le n$ 且 $1 \le x \le n$。
- $2\ p\ x$：保证 $1 \le p \le n$ 且 $1 \le x \le 10^9$。
- $3\ x\ k\ a_1\ a_2\ \ldots\ a_k$：保证 $1 \le x \le n$ 且 $1 \le a_1 < a_2 < \ldots < a_k \le n$ 且 $c_x \in \{a_1, a_2, \cdots, a_k\}$。

保证所有数据 n 之和与 q 之和均不超过 3×10^5，且所有数据 k 之和不超过 10^6。

Output

对于每次操作 3 输出一行一个整数，表示取走的球的权值之和的最大值。

Example

standard input	standard output
2 5 10 1 2 3 1 2 1 10 100 1000 10000 3 3 1 3 3 3 2 2 3 2 5 20000 2 3 200 3 3 2 1 3 3 3 3 1 2 3 1 3 4 2 1 100000 1 2 2 3 1 2 1 2 4 1 1 2 3 4 1000000 1000000 1000000 1000000 3 4 4 1 2 3 4	100 110 1200 21211 100010 4000000

Problem G. 交换操作

给定长度为 n 的非负整数序列 $A = a_1, a_2, \ldots, a_n$，定义

$$F(A) = \max_{1 \le k < n} ((a_1 \,\&\, a_2 \,\&\, \cdots \,\&\, a_k) + (a_{k+1} \,\&\, a_{k+2} \,\&\, \cdots \,\&\, a_n))$$

其中 $\&$ 表示按位与操作。

您可以进行至多一次交换操作：选择两个下标 i 和 j 满足 $1 \le i < j \le n$，交换 a_i 与 a_j 的值。

求经过至多一次交换后，$F(A)$ 的最大值。

Input

有多组测试数据。第一行输入一个整数 T 表示测试数据组数。对于每组测试数据：

第一行输入一个整数 n（$2 \le n \le 10^5$），表示序列 A 的长度。

第二行输入 n 个整数 $a_1, a_2, \cdots, a_n$（$0 \le a_i \le 10^9$），表示给定的序列 A。

保证所有数据 n 之和不超过 10^5。

Output

每组数据输出一行一个整数，表示经过至多一次交换后 $F(A)$ 的最大值。

Example

standard input	standard output
3 6 6 5 4 3 5 6 6 1 2 1 1 2 2 5 1 1 2 2 2	7 3 3

Note

对于第一组样例数据，可以交换 a_4 和 a_6 将序列变为 $\{6, 5, 4, 6, 5, 3\}$，然后选择 $k = 5$，就得到了 $F(A) = (6 \,\&\, 5 \,\&\, 4 \,\&\, 6 \,\&\, 5) + (3) = 7$。

对于第二组样例数据，可以交换 a_2 和 a_4 将序列变为 $\{1, 1, 1, 2, 2, 2\}$，然后选择 $k = 3$，就得到了 $F(A) = (1 \,\&\, 1 \,\&\, 1) + (2 \,\&\, 2 \,\&\, 2) = 3$。

对于第三组样例数据，不进行交换操作，然后选择 $k = 2$，就得到了 $F(A) = (1 \,\&\, 1) + (2 \,\&\, 2 \,\&\, 2) = 3$。

Problem H. 流画溢彩

有一个长度为 n 的序列，一开始序列中的所有元素均为 0。另外还有 m 个操作，其中第 i 个操作会将序列中第 l_i 个元素的值改为 x_i，以及将序列中第 r_i 个元素的值改为 y_i。每个操作必须恰好执行一次。

求执行操作的最优顺序，使得所有操作执行完成后，序列中所有元素之和最大。

Input

有多组测试数据。第一行输入一个整数 T 表示测试数据组数。对于每组测试数据:

第一行输入两个整数 n 和 m（$2 \le n, m \le 5 \times 10^5$）表示序列的长度和操作的个数。

对于接下来 m 行，第 i 行输入四个整数 l_i，x_i，r_i 和 y_i（$1 \le l_i < r_i \le n$，$1 \le x_i, y_i \le 2$）表示第 i 个操作。

保证所有数据 n 之和与 m 之和均不超过 5×10^5。

Output

每组数据首先输出一行一个整数，表示执行操作后，所有元素的最大和。接下来输出一行 m 个由单个空格分隔的整数 $a_1, a_2, \cdots, a_m$ 表示执行操作的最优顺序，其中 a_i 表示第 i 次执行的操作的编号。从 1 到 m 的每个整数（含两端）必须恰好出现一次。若有多种合法答案，您可以输出任意一种。

Example

standard input	standard output
2 4 4 1 1 2 2 3 2 4 1 1 2 3 2 2 1 4 1 4 2 3 2 4 1 1 2 3 1	7 4 1 3 2 5 2 1

Note

对于第一组样例数据，按 $4, 1, 3, 2$ 的顺序执行操作后，序列变为 $\{2, 2, 2, 1\}$，元素之和为 7。

对于第二组样例数据，按 $2, 1$ 的顺序执行操作后，序列变为 $\{2, 0, 2, 1\}$，元素之和为 5。

Problem I. 路径规划

有一个 n 行 m 列的网格。网格里的每个格子都写着一个整数，其中第 i 行第 j 列的格子里写着整数 $a_{i,j}$。从 0 到 $(n \times m - 1)$ 的每个整数（含两端）在网格里都恰好出现一次。

令 (i,j) 表示位于第 i 行第 j 列的格子。您现在需要从 $(1,1)$ 出发并前往 (n,m)。当您位于格子 (i,j) 时，您可以选择走到右方的格子 $(i,j+1)$（若 $j < m$），也可以选择走到下方的格子 $(i+1,j)$（若 $i < n$）。

令 $\mathbb{S}$ 表示路径上每个格子里的整数形成的集合，包括 $a_{1,1}$ 和 $a_{n,m}$。令 $\text{mex}(\mathbb{S})$ 表示不属于 $\mathbb{S}$ 的最小非负整数。请找出一条路径以最大化 $\text{mex}(\mathbb{S})$，并求出这个最大的值。

Input

有多组测试数据。第一行输入一个整数 T 表示测试数据组数。对于每组测试数据：

第一行输入两个整数 n 和 m（$1 \le n, m \le 10^6$，$1 \le n \times m \le 10^6$）表示网格的行数和列数。

对于接下来 n 行，第 i 行输入 m 个整数 $a_{i,1}, a_{i,2}, \cdots, a_{i,m}$（$0 \le a_{i,j} < n \times m$），其中 $a_{i,j}$ 表示格子 (i,j) 里的整数。从 0 到 $(n \times m - 1)$ 的每个整数（含两端）在网格里都恰好出现一次。

保证所有数据 $n \times m$ 之和不超过 10^6。

Output

每组数据输出一行一个整数，表示最大的 $\text{mex}(\mathbb{S})$。

Example

standard input	standard output
2 2 3 1 2 4 3 0 5 1 5 1 3 0 4 2	3 5

Note

对于第一组样例数据，共有 3 条可能的路径。

- 第一条路径为 $(1,1) \to (1,2) \to (1,3) \to (2,3)$。$\mathbb{S} = \{1,2,4,5\}$ 因此 $\text{mex}(\mathbb{S}) = 0$。
- 第二条路径为 $(1,1) \to (1,2) \to (2,2) \to (2,3)$。$\mathbb{S} = \{1,2,0,5\}$ 因此 $\text{mex}(\mathbb{S}) = 3$。
- 第三条路径为 $(1,1) \to (2,1) \to (2,2) \to (2,3)$。$\mathbb{S} = \{1,3,0,5\}$ 因此 $\text{mex}(\mathbb{S}) = 2$。

因此答案为 3。

对于第二组样例数据，只有 1 条可能的路径，即 $(1,1) \to (1,2) \to (1,3) \to (1,4) \to (1,5)$。$\mathbb{S} = \{1,3,0,4,2\}$ 因此 $\text{mex}(\mathbb{S}) = 5$。

Problem J. X 等于 Y

对于正整数 X 与 $b \geq 2$，定义 $f(X, b)$ 为一个描述了 X 在 b 进制表示下的序列，其中序列的第 i 个元素表示 X 在 b 进制表示下从低到高第 i 位的值。例如，$f(6, 2) = \{0, 1, 1\}$，而 $f(233, 17) = \{12, 13\}$。

给定的四个正整数 x，y，A 和 B，请找到两个正整数 a 和 b，同时满足：

- $2 \leq a \leq A$
- $2 \leq b \leq B$
- $f(x, a) = f(y, b)$

Input

有多组测试数据。第一行输入一个整数 T（$1 \leq T \leq 10^3$）表示测试数据组数。对于每组测试数据：

第一行输入四个整数 x，y，A 和 B（$1 \leq x, y \leq 10^9$，$2 \leq A, B \leq 10^9$）。

保证至多只有 50 组测试数据满足 $\max(x, y) > 10^6$。

Output

对于每组测试数据，如果不存在合法的正整数 a 和 b，则输出一行一个字符串 `NO`。

否则，首先输出一行一个字符串 `YES`。下一行输出两个由单个空格分隔的整数 a 和 b。如果有多种合法答案，您可以输出任意一种。

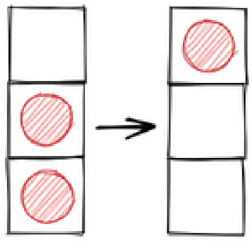
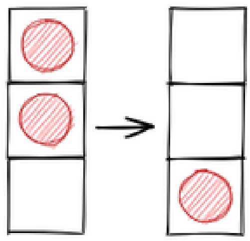
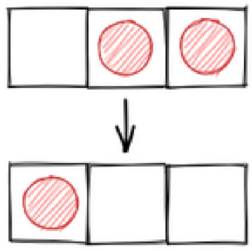
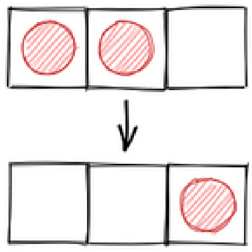
Example

standard input	standard output
6 1 1 1000 1000 1 2 1000 1000 3 11 1000 1000 157 291 5 6 157 291 3 6 10126 114514 789 12345	YES 2 2 NO YES 2 10 YES 4 5 NO YES 779 9478

Problem K. 独立钻石

“独立钻石”是一种单人桌游。游戏在 n 行 m 列的棋盘上进行，棋盘上的每一格要么是空格，要么有一枚棋子。一开始，棋盘上共有 k 枚棋子。

在游戏中，玩家可以选择一枚棋子，将它跳过相邻棋子到空格上，并移除被跳过的棋子。具体来说，令 (i, j) 表示位于第 i 行第 j 列的格子，玩家可以执行以下四种操作。

操作	描述	图例
向上跳	选择 (i, j) 同时满足 • $i \ge 3$。 • (i, j) 和 $(i-1, j)$ 中各有一枚棋子。 • $(i-2, j)$ 是空格。 将 (i, j) 中的棋子跳到 $(i-2, j)$ 中，并移除 $(i-1, j)$ 中的棋子。	
向下跳	选择 (i, j) 同时满足 • $i \le n-2$。 • (i, j) 和 $(i+1, j)$ 中各有一枚棋子。 • $(i+2, j)$ 是空格。 将 (i, j) 中的棋子跳到 $(i+2, j)$ 中，并移除 $(i+1, j)$ 中的棋子。	
向左跳	选择 (i, j) 同时满足 • $j \ge 3$。 • (i, j) 和 $(i, j-1)$ 中各有一枚棋子。 • $(i, j-2)$ 是空格。 将 (i, j) 中的棋子跳到 $(i, j-2)$ 中，并移除 $(i, j-1)$ 中的棋子。	
向右跳	选择 (i, j) 同时满足 • $j \le m-2$。 • (i, j) 和 $(i, j+1)$ 中各有一枚棋子。 • $(i, j+2)$ 是空格。 将 (i, j) 中的棋子跳到 $(i, j+2)$ 中，并移除 $(i, j+1)$ 中的棋子。	

给定一个初始的棋盘，求经过任意次操作（包括零次）之后，棋盘上最少能剩余几枚棋子。

Input

有多组测试数据。第一行输入一个整数 T（$1 \le T \le 20$）表示测试数据组数。对于每组测试数据：

第一行输入三个整数 n，m 和 k（$1 \le n, m \le 6$，$1 \le k \le \min(6, n \times m)$）表示棋盘的行数，列数和初始棋子的数量。

对于接下来 k 行，第 i 行输入两个整数 x_i 和 y_i（$1 \le x_i \le n$，$1 \le y_i \le m$）表示一开始第 x_i 行第 y_i 列的格子里有一枚棋子。除了这 k 个格子之外，其它格子一开始都是空格。这 k 个格子的位置不会重复。

Output

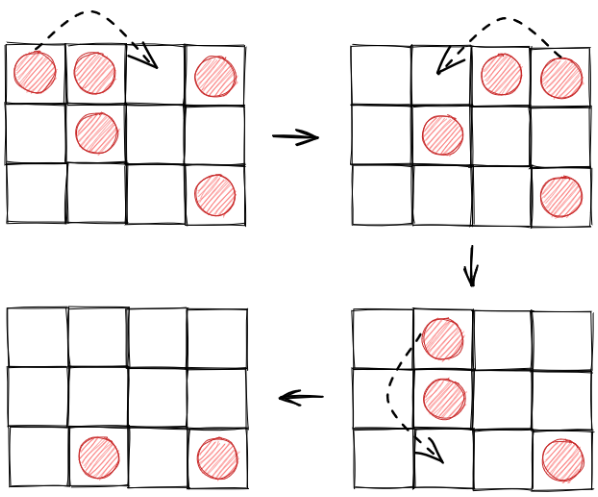
每组数据输出一行一个整数，表示经过任意次操作（包括零次）之后，棋盘上最少能剩余几枚棋子。

Example

standard input	standard output
3 3 4 5 2 2 1 2 1 4 3 4 1 1 1 3 3 1 1 1 2 1 3 2 1 1 2 1	2 3 1

Note

第一组样例数据解释如下。



对于第二组样例数据，由于初始棋盘不存在空格，因此无法进行任何操作。

对于第三组样例数据，由于棋盘不足三格，因此无法进行任何操作。

Problem L. 经典问题

给定一张 n 个点的无向完全图与 m 个三元组 $P_1, P_2, \cdots, P_m$，其中 $P_i = (u_i, v_i, w_i)$。保证 $1 \le u_i < v_i \le n$，且对于任意两个编号不同的三元组 P_i 和 P_j，有 $(u_i, v_i) \ne (u_j, v_j)$。

对于图中的任意两个节点 x 与 y（$1 \le x < y \le n$），定义它们之间的无向边的边权如下：

- 如果存在一个三元组 P_i 满足 $u_i = x$ 且 $v_i = y$，那么边权为 w_i。
- 否则，边权为 $|x - y|$。

求这张图的最小生成树的边权之和。

Input

有多组测试数据。第一行输入一个整数 T（$1 \le T \le 10^5$）表示测试数据组数。对于每组测试数据：

第一行输入两个整数 n 和 m（$1 \le n \le 10^9$，$0 \le m \le 10^5$）表示图的点数与三元组的数量。

对于接下来 m 行，第 i 行输入三个整数 u_i，v_i 和 w_i（$1 \le u_i < v_i \le n$，$0 \le w_i \le 10^9$）表示第 i 个三元组。保证对于所有 $1 \le i < j \le m$ 都有 $(u_i, v_i) \ne (u_j, v_j)$。

保证所有数据 m 之和不超过 5×10^5。

Output

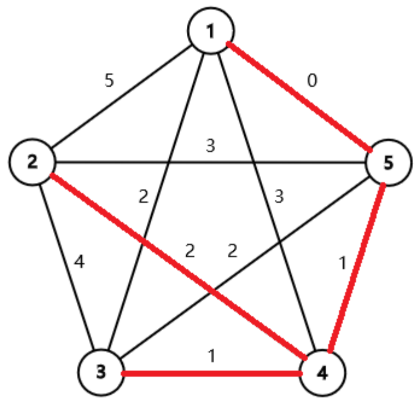
每组数据输出一行一个整数，表示这张图的最小生成树的边权之和。

Example

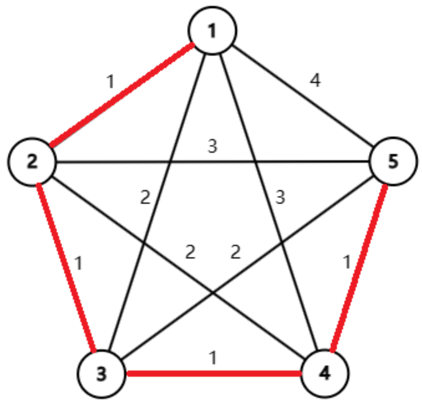
standard input	standard output
3 5 3 1 2 5 2 3 4 1 5 0 5 0 5 4 1 2 1000000000 1 3 1000000000 1 4 1000000000 1 5 1000000000	4 4 1000000003

Note

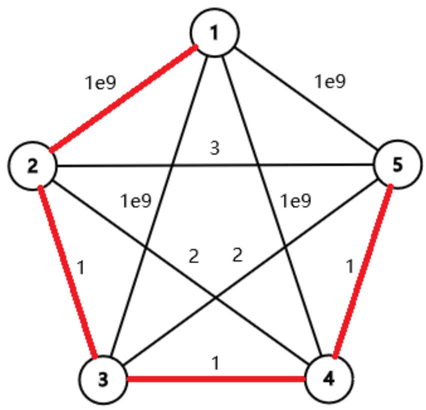
第一组样例数据如下图所示，最小生成树用红色线段标出。



第二组样例数据如下图所示，最小生成树用红色线段标出。



第三组样例数据如下图所示，最小生成树用红色线段标出。



Problem M. 计算几何

给定一个有 n 个顶点的凸多边形 P，您需要选择 P 的两个顶点，并用一条同时穿过这两个顶点的直线，将 P 分成两个面积均为正数的小多边形 Q 和 R。

记 $d(Q)$ 表示多边形 Q 的直径，$d(R)$ 表示多边形 R 的直径，求 $(d(Q))^2 + (d(R))^2$ 的最小值。

请回忆：一个多边形的直径，指的是该多边形内部或边界上任意两点之间的距离的最大值。

Input

有多组测试数据。第一行输入一个整数 T 表示测试数据组数。对于每组测试数据：

第一行输入一个整数 n（$4 \le n \le 5 \times 10^3$）表示凸多边形 P 的顶点数量。

对于接下来 n 行，第 i 行输入两个整数 x_i 和 y_i（$0 \le x_i, y_i \le 10^9$），表示凸多边形 P 的第 i 个顶点。顶点按逆时针顺序给出。保证该凸多边形面积为正，且没有顶点会重合。可能存在三个顶点位于同一条直线上的情况。

保证所有数据 n 之和不超过 5×10^3。

Output

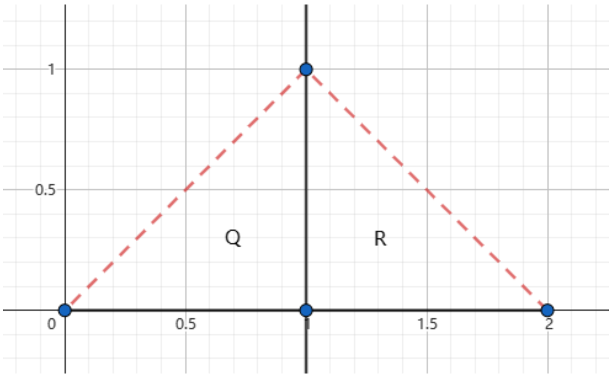
每组数据输出一行一个整数表示答案。

Example

standard input	standard output
2 4 1 0 2 0 1 1 0 0 6 10 4 9 7 5 7 4 5 6 4 9 3	4 44

Note

第一组样例数据如下图所示。小多边形的直径用红色虚线标出。事实上，顶点 $(1,0)$ 和 $(1,1)$ 是这一组数据中唯一能选择的一对顶点。您不能选择顶点 $(0,0)$ 和 $(2,0)$，或顶点 $(0,0)$ 和 $(1,1)$，因为它们无法将 P 分成两个面积均为正数的小多边形。



第二组样例数据如下图所示。小多边形的直径用红色虚线标出。