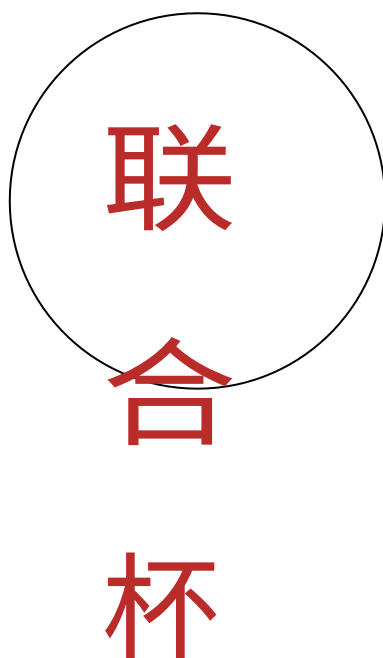


# 第二届环球杯



## 第 11 赛段：南京

2023 年 11 月 25-26 日

该问题集应包含 13 个问题，共 28 页。

根据



国际大学生程序设计竞赛 (ICPC)

主办方

筹备



# 问题 A.酷，它是昨天的四倍<sup>25-26 日</sup>

输入文件：标准输入  
输出文件：标准输出  
时间限制 1 秒 内存限制  
1024 兆字节

继2018年、2019年、2020年、2021年和2022年的成功举办后，南京航空航天大学将连续第六次承办国际大学生程序设计竞赛（ICPC）南京赛区的比赛。

2018年和2019年，清华大学的*Power of Two*队和*Three Hold Two*队获得冠军。2020年、2021年和2022年，北京大学*倒十字*队获得三连冠。今年，共有约 330 支队伍参赛。最多将产生 33 枚金牌、66 枚银牌和 99 枚铜牌（注：这些数字仅供参考）。我们期待看到参赛者的出色表现！

更令人高兴的是，随着疫情的结束，我们终于可以齐聚南京，参加这次精彩的比赛。我们要感谢所有工作人员和志愿者为这次比赛付出的辛勤劳动。感谢你们为本次竞赛做出的巨大贡献！



2018 ICPC 亚洲南京地区竞赛

在 2018 年的竞赛中，问题 K "*袋鼠之谜*"要求参赛者游戏构建一个操作序列：

25-26 日

谜题是一个有  $n$  行和  $m$  列 ( $1 \leq n, m \leq 20$ ) 的网格，网格中站着几只（至少 2 只）袋鼠。玩家的目標是控制它們走到一起。有些格子里有墙，袋鼠不能进入有墙的格子。其他牢房则是空的。袋鼠可以从一个空格向相邻的空格移动，移动的方向有四个：上、下、左、右。

一开始，每个空格中都有一只袋鼠，玩家可以通过按键盘上的 U、D、L、R 按钮来控制袋鼠。袋鼠会根据您按下的按钮同时移动。

参赛者需要构建一个最多由  $5 \times 10^4$  步组成的操作序列，其中只包括 U、D、L、R，以实现目标。

在 2020 年的竞赛中，问题 A “**啊，昨日重现**”<sup>25-26 日</sup>要求参赛者构建一个输入地图，以破解前述问题的以下代码：

```
#include <bits/stdc++.h>
using namespace std;
; strings = "UDLR"
; int main ()
{
    srand ( time (NULL));
    for ( int i = 1 ; i <= 50000 ; i++) putchar ( s [ rand () % 4 ] );
    返回 0 ;
}
```

在 2021 年的竞赛中，问题 A “**哎呀，又是昨天了**”也要求参赛者游戏构建一个操作序列：

这一次，网格中的每个单元格都正好代表一只袋鼠。您需要构建一个仅由字符 "U"、"D"、"L" 和 "R" 组成的操作序列。应用该操作序列后，您必须确保每只袋鼠都会聚集到特定的单元格 ( $a, b$ )。操作序列的长度不能超过  $3(n - 1)$ 。与往常一样，袋鼠会根据您的指令同时移动。

在 2022 年的竞赛中，问题 A “**停止，昨天请不要再继续了**”要求参赛者解决下面的计算问题：

这一次，网格中的每个单元格（除了一个洞）都正好站着一只袋鼠。操作顺序已给出，所有走出网格或踏上洞口的袋鼠都将被移走。给定所有操作后剩余袋鼠的数量，计算可能是洞的位置数。

现在，在 2023 年的比赛中，袋鼠问题再次出现！我们不知道问题设置者为什么对袋鼠如此着迷，但问题如下：

给你一个  $n$  行  $m$  列的网格。每个单元格要么是洞，要么是空的。每个空格中正好有一只袋鼠。

同样，按下键盘上的 U、D、L、R 按钮即可控制袋鼠。所有袋鼠将根据按下的按钮同时移动。具体来说，对于位于第  $i$  行第  $j$  列单元格中的袋鼠，用  $(i, j)$  表示：

1. 按钮 U：它将移动到  $(i - 1, j)$ 。
2. 按钮 D：将移动到  $(i + 1, j)$ 。
3. 按钮 L：将移动到  $(i, j - 1)$ 。
4. 按钮 R：它将移动到  $(i, j + 1)$ 。

如果一只袋鼠踩到洞上或走出网格，它就会被从网格中移除。如果在进行一系列操作（可能是空操作）后，网格上只剩下一只小袋鼠，那么这只小袋鼠就是获胜者。



问题是：对于每只袋鼠，确定是否存在使其成为赢家的操作序列。输出可能成为赢家的袋鼠总数。

输入

有多个测试用例。输入的第一行包含一个整数  $T$ ，表示测试用例的数量。对于每个测试用例

第一行包含两个整数  $n$  和  $m$  ( $1 \leq n, m \leq 10^3$ ;  $1 \leq n \times m \leq 10^3$ )，表示网格的行数和列数。

对于下面的  $n$  行，第  $i$  行包含长度为  $m$  的字符串  $s_{i,1}s_{i,2} \dots s_{i,m}$ ，其中每个字符都是 '.' (点，ascii: 46) 或 'O' (大写字母，ascii: 79)。如果  $s_{i,j}$  是 '.'，则网格  $(i, j)$  为空；如果  $s_{i,j}$  是 'O'，则网格  $(i, j)$  为洞。

保证所有测试用例的  $n \times m$  之和不超过  $5 \times 10^3$ 。

输出

对每个测试用例输出一行，其中包含一个整数，表示有多少只袋鼠的操作序列能使其获胜。

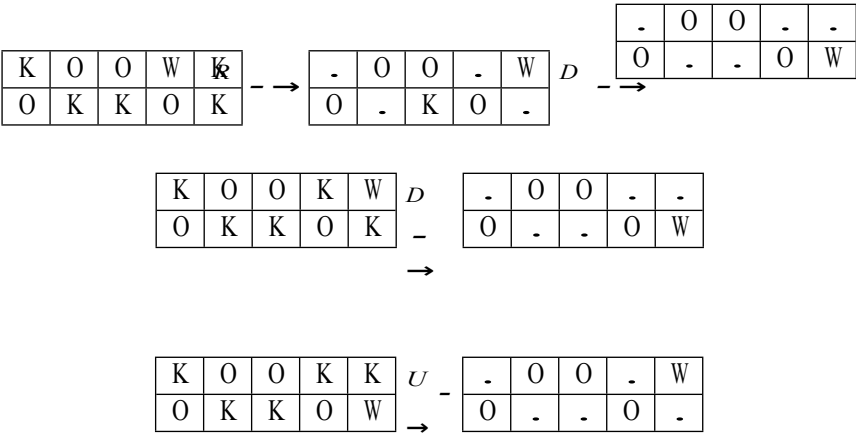
示例

标准输入	标准输出
4	3
2 5	1
.OO.....	0
O...O.	0
1 3	
O.O	
1 3	
.O.	
2 3	
000	
000	

备注

测试案例示例说明如下。我们用 "W" 表示后来成为赢家的袋鼠，用 "K" 表示其他袋鼠。

在第一个样本测试案例中，最初位于 (1, 4)、(1, 5) 和 (2, 5) 的袋鼠可能成为赢家。可能的操作顺序如下：



对于第二个样本测试案例，由于只有一只袋鼠，因此无需进行任何操作即可成为获胜者。

对于第三个样本测试案例，由于任何操作都会同时移除两只袋鼠，因此不可能有赢家。

在第四个样本测试案例中，由于没有袋鼠，所以不可能有赢家。





问题 B. 联合路交叉口

输入文件：	标准输入
输出文件：	标准输出
时间限制	1.5 秒 内存限
制	1024 兆字节

交集大于联合 (*Intersection over Union*)，又称雅卡指数和雅卡相似系数（最初由保罗-雅卡（Paul Jaccard）命名，法文名称为 *coefficient de coïncidence*），是一种用于衡量样本集相似性和多样性的统计量。雅卡系数用于衡量有限样本集之间的相似性，其定义为样本集的交集大小除以联合集的大小：

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

交集大于联合（IOU）也是计算机视觉中广泛使用的一种指标，用于评估物体检测和分割算法。与 x 轴和 y 轴对齐的矩形（即其边与 x 轴和 y 轴平行）通常称为 "轴对齐矩形"、"轴对齐包围盒"（AABB）或简称 "包围盒"。另一方面，不一定与 x 轴和 y 轴对齐的矩形（即其边可能成一定角度）通常被称为 "旋转矩形"、"旋转边界框" 或 "定向边界框"（OBB）。在计算机视觉和图像处理应用中，这两种矩形都被广泛使用，具体取决于手头的问题。

在这个问题中，你的任务是找到一个轴对齐的矩形 (AABB)，使其与旋转矩形 (OBB) 的欠最大。两个矩形之间的 IOU 定义为两个矩形的相交面积除以它们的结合面积。

输入

有多个测试用例。输入的第一行包含一个整数  $T$  ( $1 \leq T \leq 10^4$ )，表示测试用例的数量。对于每个测试用例

第一行也是唯一一行包含八个整数  $x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4$  ( $-10^9 \leq x_i, y_i \leq 10^9$ )，其中  $(x_i, y_i)$  表示旋转矩形第  $i$  个顶点的坐标，按顺时针或逆时针顺序排列。可以保证旋转矩形的面积为正值。

输出

对于每个测试用例，输出一行，其中包含一个数字，表示旋转矩形与轴对齐矩形之间的最大 IOU。如果绝对或相对误差不超过  $10^{-9}$ ，您的答案将被视为正确。

示例

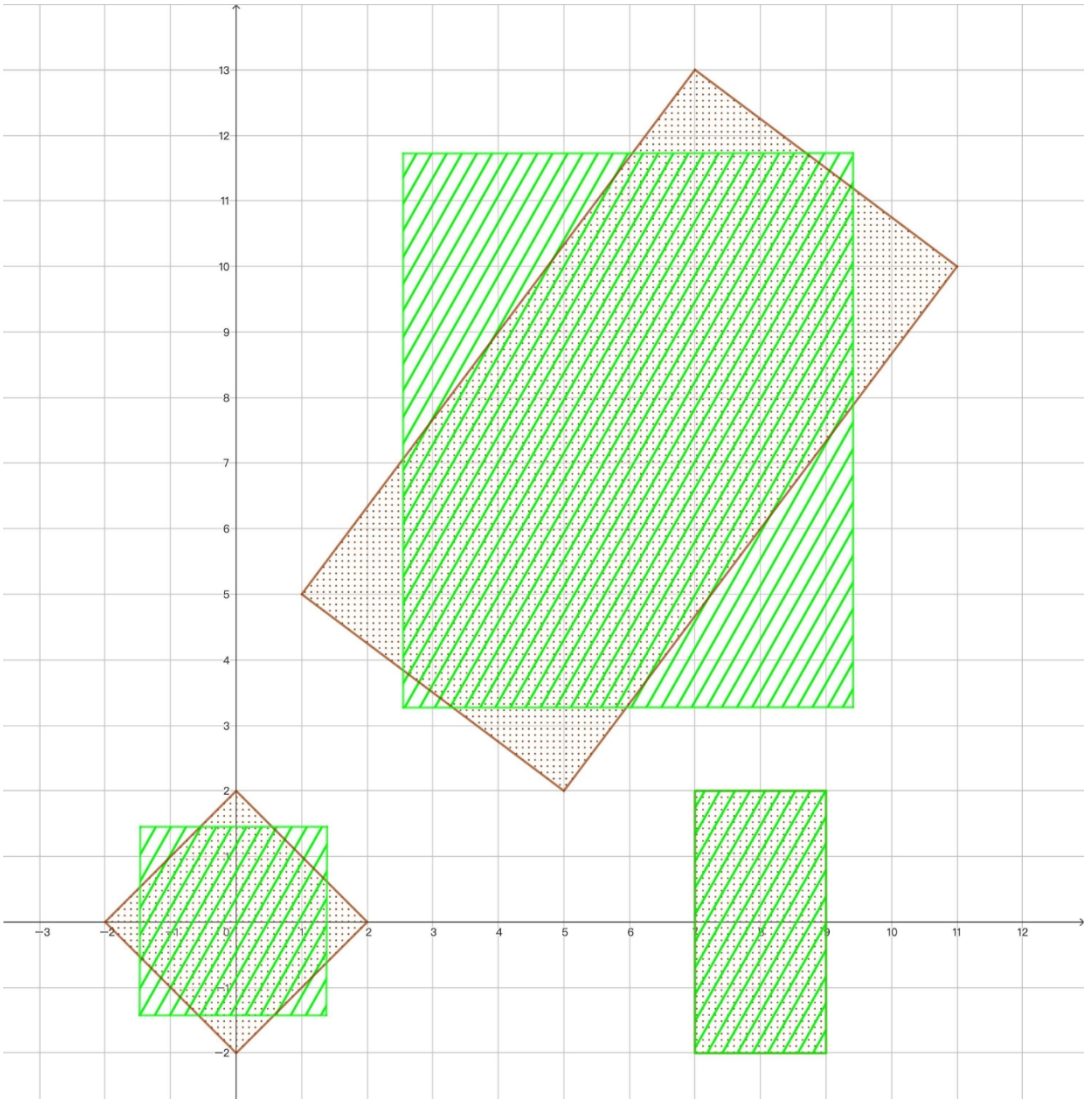
标准输入	标准输出
------	------



3 0 2 2 0 0 -2 -2 0 7 -2 9 -2 9 2 7 2 7 13 11 10 5 2 1 5	0.70710678118654752 1 0.62384322483109367
---	---

备注

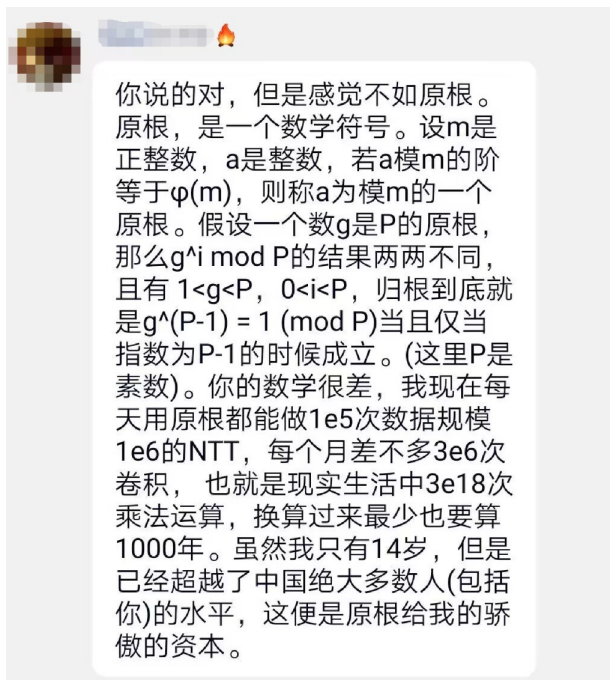
测试案例示例如下。输入的矩形用点表示，最佳轴对齐矩形用填充表示。



## 问题 c. 原始根

输入文件：标准输入  
输出文件：标准输出  
时间限制 1 秒 内存限制  
1024 兆字节

包包刚刚学习了数论中的原始根，正在通过即时通讯软件向小青鱼炫耀他的知识。



本图片仅供娱乐，与问题本身无关。如果您不懂中文，可以放心跳过此图。

如果一个非负整数  $g$  是  $P$  的基元根模数（其中  $P$  是一个素数），那么  $g^{P-1} \equiv 1 \pmod{P}$ ，基于这一事实，包包决定使用表达式  $(g^{(P-1)}) \% P == 1$  来检查  $g$  是否是  $P$  的基元根模数。不幸的是，在大多数编程语言（例如 C 和 C++）中， $\wedge$  是位排他运算符 (XOR)，而不是幂运算符。小青鱼一眼就发现了这个问题，现在他对下面的问题很感兴趣：

给定一个质数  $P$  和一个非负整数  $m$ ，有多少个非负整数  $g$  满足  $g \leq m$

且  $g \otimes (P-1) \equiv 1 \pmod{P}$ ? 这里的  $\otimes$  是比特排他或 (XOR) 运算符。请

帮助小青鱼解决这个问题。

## 输入

有多个测试用例。输入的第一行包含一个整数  $T$  ( $1 \leq T \leq 10^5$ )，表示测试用例的数量。对于每个测试用例



第一行也是唯一一行包含两个整数  $P$  和  $m$  ( $25 \leq P \leq 10^{18}$ ,  $0 \leq m \leq 10^{18}$ ,  $P$  是质数)。

## 输出

对于每个测试用例，输出一行，其中包含一个整数，表示满足约束条件的  $g$  的数量。

标准输入	标准输出
3 2 0 7 11 1145141 998244353	1 2 872

## 问题 D. 红黑树

输入文件：        标准输入  
输出文件：        标准输出  
时间限制          2 秒 内存限制  
                    1024 兆字节

有一棵有根的树，树上有  $n$  个顶点，编号从 1 到  $n$ ，其中顶点 1 是树根。每个顶点都有一种颜色，要么是红色，要么是黑色。

如果从一个顶点到它的任何子叶顶点的每一条简单路径都包含相同数量的黑色顶点，我们就说这个顶点是好的。如果一棵树的所有顶点都是好的，我们就说这棵树是完美的。

让  $R_k$  是根植于顶点  $k$  的子树。对于每个  $1 \leq k \leq n$ ，请回答下面的问题：如果可以选择一组顶点并改变它们的颜色（即把红色顶点改成黑色，把黑色顶点改成红色），请计算要使  $R_k$  完美，至少需要选择多少个顶点。

回顾一下，简单路径是指不多次经过同一条边的路径。

另请注意，以顶点  $k$  为根的子树是一棵由顶点  $k$  的所有后代组成的树，并以顶点  $k$  为根。请注意，任何顶点都是其自身的后代。

## 输入

有多个测试用例。输入的第一行包含一个整数  $T$ ，表示测试用例的数量。对于每个测试用例

第一行包含一个整数  $n$  ( $2 \leq n \leq 10^5$ )，表示树的顶点数。

第二行包含长度为  $n$  的字符串  $s_{s_{12}} \dots s_n$  ( $s_i \in \{0, 1\}$ )。如果  $s_i = 0$ ，则顶点  $i$  为红色；如果  $s_i = 1$ ，则顶点  $i$  为黑色。

第三行包含  $(n - 1)$  个整数  $p_2, p_3, \dots, p_n$  ( $1 \leq p_i < i$ )，其中  $p_i$  是顶点  $i$  的父节点。保证所有测试用例的  $n$  之和不超过  $10^6$ 。

## 输出

对于每个测试用例，输出一行包含  $n$  个用空格分隔的整数，其中第  $i$  个整数表示为使  $R_i$  完美而必须更改颜色的顶点的最少数目。

请不要在每行末尾输出额外的空格，否则您的解决方案可能会被视为错误！

## 示例

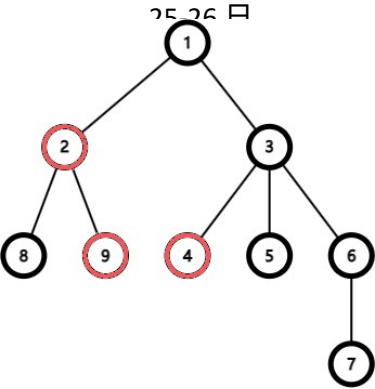
标准输入	标准输出
------	------



2子	4 1 2 0 0 0 0 0
9	25-26日
101011110	2 0 0 0
1 1 3 3 3     6 2 2	
4	
1011	
1 1 3	

备注





我们对第一个示例测试用例进行说明。

为了让  $R_1$  更完美，我们可以改变顶点 2、4、6 和 9 的颜色。这样，从顶点 1 到它的后代叶子的所有路径都将包含 3 个黑色顶点，从顶点 2 到它的后代叶子的所有路径都将包含 2 个黑色顶点，从顶点 3 到它的后代叶子的所有路径都将包含 2 个黑色顶点。由于顶点 4 至 9 只有一片后代叶，因此它们总是好的。

为了使  $R_2$  完美，我们可以改变顶点 8 的颜色。之后，从顶点 2 到其后代叶子的所有路径都将包含 0 个顶点。由于顶点 8 和 9 只有一片后代叶，因此它们总是好的。

## 问题 E. 扩展距离

输入文件：标准输入  
输出文件：标准输出  
时间限制：1 秒 内存限制：1024 兆字节

给定  $n \times m$  个排列成  $n$  行和  $m$  列的点，相邻点之间存在加权双向边。也就是说，设  $(i, j)$  是第  $i$  行和第  $j$  列上的点，对于所有  $1 \leq i, i' \leq n$  和  $1 \leq j, j' \leq m$ ，当且仅当  $|i - i'| + |j - j'| = 1$  时， $(i, j)$  和  $(i', j')$  之间有一条边。

小宝将从第一列的任意一点  $(p, 1)$  开始旅行，在最后一列的任意一点  $(q, m)$  结束旅行。他可以沿边缘双向行进。路径的距离定义为路径中各条边的权重之和。在从第一列到最后一列的所有路径中，小宝会选择最短的路径。

小青鱼希望包包能更好地享受旅途，所以他尝试在包包开始旅行之前增加一些边的重量。具体来说，"小青鱼"可以通过一次操作将任意边的权重增加 1。他希望经过所有操作后，小宝宝的路径距离能增加  $k$ 。请帮他计算所需的最小操作次数，并输出相应的解。

### 输入

有多个测试用例。输入的第一行包含一个整数  $T$ ，表示测试用例的数量。对于每个测试用例

第一行包含三个整数  $n$ 、 $m$  和  $k$  ( $1 \leq n \times m \leq 500$ ,  $2 \leq n, m \leq 500$ ,  $1 \leq k \leq 100$ )，表示行数和列数，以及最短路径长度的增量。

对于以下  $n$  行，第  $i$  行包含  $(m - 1)$  个整数  $r_{i,1}, r_{i,2}, \dots, r_{i,m-1}$  ( $1 \leq r_{i,j} \leq 10^9$ )，其中  $r_{i,j}$  表示  $(i, j)$  和  $(i, j + 1)$  之间边的权重。

对于以下  $(n - 1)$  行，第  $i$  行包含  $m$  个整数  $c_{i,1}, c_{i,2}, \dots, c_{i,m}$  ( $1 \leq c_{i,j} \leq 10^9$ )，其中  $c_{i,j}$  表示  $(i, j)$  和  $(i + 1, j)$  之间的边的权重。

保证所有测试用例的  $n \times m$  之和不超过  $5 \times 10^3$ 。

### 输出

对于每个测试用例

首先输出一行，其中包含一个整数，表示所需的最小操作数。

然后输出  $n$  行。第  $i$  行包含  $(m - 1)$  个整数  $r_{i,1}, r_{i,2}, \dots, r_{i,m-1}$  ( $1 \leq r_{i,j} \leq 2 \times 10^9$ )

用空格隔开，其中  $r_{i,j}$  表示  $(i, j)$  和  $(i, j + 1)$  之间的边的权重。

业务。然后输出  $(n - 1)$  行。第  $i$  行包含  $m$

个整数  $c^r$

$, c_{i,2}^r, \dots, c_{i,j}^r \quad (1 \leq i, j \leq 2 \times 10^9)$  分离

空格，其中  $c^r$  表示所有操作后  $(i, j)$  和  $(i+1, j)$  之间的边的权重。

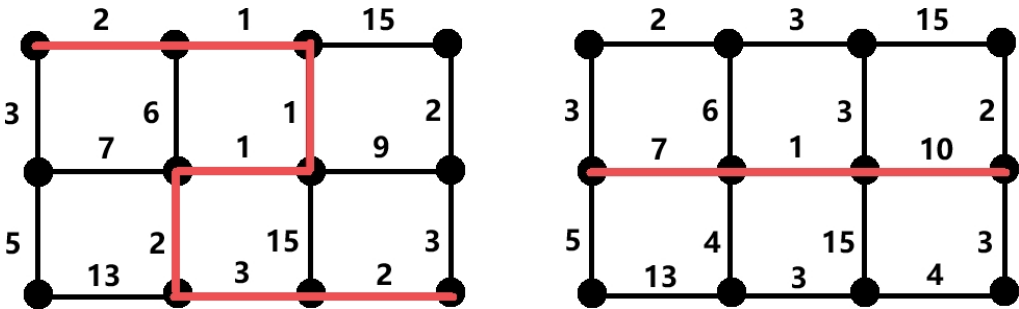
如果有多个有效答案，则输出其中任何一个。

请不要在每行末尾输出额外的空格，否则您的解决方案可能会被视为错误！

标准输入	标准输出
2	9
3 4 6	2 3 15
2 1 15	7 1 10
7 1 9	13 3 4
13 3 2	3 6 3 2
3 6 1 2	5 4 15 3
5 2 15 3	4
3 3 3	4 1
1 1	3 2
2 2	3 3
3 3	1 1 1
1 1 1	2 2 2
2 2 2	

备注

第一个测试案例示例如下。左边的图是原始图，右边的图是增加某些边的权重后的图。每个图的最短路径用红线标出。



## 问题 F. 等价改写

输入文件：标准输入  
输出文件：标准输出  
时间限制：2 秒 内存限制：1024 兆字节

有一个长度为  $m$  的序列  $A$ ，其中所有元素都等于 0。我们将依次对  $A$  执行  $n$  次操作。第  $i$  次操作可以用  $p_i$  个不同的整数  $b_{i,1}, b_{i,2}, \dots, b_{i,p_i}$  表示，我们将把序列中第  $b_{i,j}$  个元素的值改为  $i$ ，所有  $1 \leq j \leq p_i$ 。让  $R$  成为所有操作后的序列。

现在我们要求您重新排列运算，但仍能得到相同的结果。更正式地说，设  $q_1, q_2, \dots, q_n$  是  $n$  与  $1, 2, \dots, n$  不同的排列组合。你将在序列  $A$  上依次执行  $q_1$ -th,  $q_2$ -th, ...,  $q_n$ -th 操作，最后得到的序列必须等于  $R$ 。你的任务是找到这样的排列组合，或者说它不存在。

回顾一下， $n$  的排列是长度为  $n$  的序列，其中从 1 到  $n$ （均包括在内）的每个整数都正好出现一次。设  $x_1, x_2, \dots, x_n$  和  $y_1, y_2, \dots, y_n$  是  $n$  的两种排列。如果存在一个整数  $k$ ，使得  $1 \leq k \leq n$  且  $x_k \neq y_k$ ，我们就说它们是不同的。

## 输入

有多个测试用例。输入的第一行包含一个整数  $T$ ，表示测试用例的数量。对于每个测试用例

第一行包含两个整数  $n$  和  $m$  ( $1 \leq n, m \leq 10^5$ )，表示操作次数和序列长度。

对于下面的  $n$  行，第  $i$  行首先包含一个整数  $p_i$  ( $1 \leq p_i \leq m$ )，表示第  $i$  次操作所改变的元素数。接着是  $p_i$  个不同的整数  $b_{i,1}, b_{i,2}, \dots, b_{i,p_i}$  ( $1 \leq b_{i,j} \leq m$ )，表示要改变的元素的索引。

保证所有测试用例的  $(n + m)$  之和不超过  $2 \times 10^6$ ，所有测试用例的  $p_i$  之和不超过 10。<sup>6</sup>

## 输出

对于每个测试用例，如果存在这种排列，首先在一行中输出 "是"。然后在第二行输出  $n$  个整数  $q_1, q_2, \dots, q_n$ ，中间用空格隔开，表示答案。如果有多个有效答案，可以输出其中任何一个。

如果没有这种排列，只需在一行中输出 "否"。

请不要在每行末尾输出额外的空格，否则您的解决方案可能会被视为错误！

## 示例

标准输入	标准输出
3	是
3 6	3 1 2



3 3 1 5	没有
2 5 3	25-26 日
2 2 6	没有
2 3	
3 1 3 2	
2 3 1	
1 3	
2 2 1	

对于第一个示例测试用例，按照 $\{1, 2, 3\}$ 或 $\{3, 1, 2\}$ 的顺序执行操作都会产生相同的结果序列 $\{1, 3, 2, 0, 2, 3\}$ 。

## 问题 G. Knapsack

输入文件：        标准输入  
输出文件：        标准输出  
时间限制          1 秒 内存限制  
                    1024 兆字节

小青鱼是个没有经验的商人，最近开了一家名为 "皇后有机珠宝" 的商店。这家珠宝店里有  $n$  颗宝石，其中第  $i$  颗宝石的价格是  $w_i$  美元，其美丽程度为

$v_i$ 。在去商店之前，你手头有  $W$  美元，打算用这笔钱购买总美度最高的宝石。

有趣的是，小青鱼的商店今天正在促销。任何到店的顾客都可以挑选任意  $k$  颗宝石并免费带回家！有了这个机会，你一定很想知道，如果你采取了最佳策略，你用  $W$  美元能获得的宝石总美貌度有多高。

请注意，商店只储存每种宝石的一个单位，因此您无法多次获得同一种宝石。另外请注意，您不必花光所有的钱。

## 输入

每个测试文件中只有一个测试用例。

输入的第一行包含三个整数  $n$ 、 $W$  和  $k$  ( $1 \leq n \leq 5 \times 10^3$ ,  $1 \leq W \leq 10^4$ ,  $0 \leq k \leq n$ )，表示商店中宝石的总数、您拥有的金额以及您可以免费拿走的宝石数量。

对于下面的  $n$  行，第  $i$  行包含两个整数  $w_i$  和  $v_i$  ( $1 \leq w_i \leq W$ ,  $1 \leq v_i \leq 10^9$ )，表示第  $i$  颗宝石的价格和美观程度。

## 输出

输出一行包含一个整数的答案。

## 实例

标准输入	标准输出
4 10 1 9 10 10 1 3 5 5 20	35
5 13 2 5 16 5 28 7 44 8 15	129





8 月 1 日	25-26 日
---------	---------

备注

在第一个例子中，小青鱼的商店里有 4 颗宝石，您可以免费获得 1 颗宝石。最佳策略是免费获得第一颗宝石，然后购买第三颗和第四颗宝石。

宝石	价格 $w_i$	美容 $v_i$	行动
1	9	10	免费获取
2	10	1	/
3	3	5	购买
4	5	20	购买

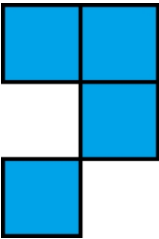
所以答案是  $10 + 5 + 20 = 35$ 。

# 问题 H. 谜题问号

25-26 日

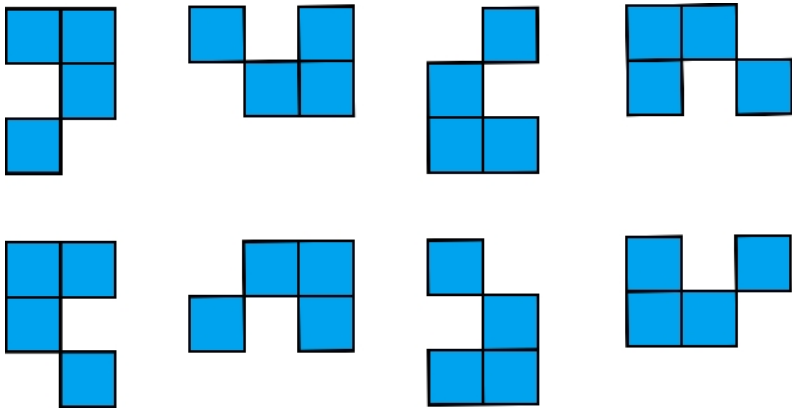
输入文件：	标准输入
输出文件：	标准输出
时间限制	3 秒 内存限制
	1024 兆字节

星慧是一位谜题高手。今天，她玩的是一道名为 "填问号" 的谜题。谜题包含一个  $n$  行  $n$  列的网格，以及一些问号填空（QM 填空）。如下图所示，一个问号填满了 4 个单元格。



一件 QM 作品（我猜它看起来像个问号？）

QM 棋子必须完全放置在网格内，棋子可以旋转 90 度和/或翻转。更确切地说，共有 8 种 QM 棋子，如下图所示。



任何两块奇数棋子都不能占据同一个格子。谜题的目标是在  $m$  的网格上找到放置最多数量的 QM 棋子的方法。

星辉想知道您能否成功解开谜题。

## 输入



有多个测试用例。输入的第一行包含一个整数  $25,264$ ，表示测试用例的数量。对于每个测试用例第一行也是唯一一行包含一个整数  $n$  ( $1 \leq n \leq 2 \times 10^3$ )，表示网格的大小。保证所有测试用例的  $n^2$  之和不超过  $5 \times 10^6$ 。

## 输出

对于每个测试用例

首先输出一行，其中包含一个整数，表示网格上放置的最大 QM 件数。

然后输出  $n$  行。每行包含  $n$  个整数，中间用空格隔开。第  $i$  行的第  $j$  个整数  $a_{i,j}$  表示第  $i$  行第  $j$  列上的单元格属于  $a_{i,j}$ -th QM piece。如果  $a_{i,j}$  为 0，则表示相应的单元格为空，不属于任何 QM 件。

如果有多个解决方案，则输出其中任何一个。

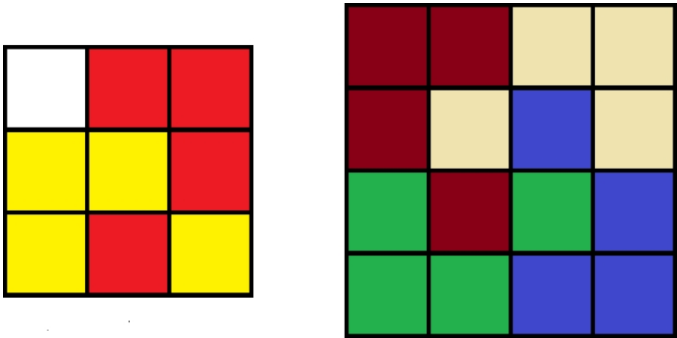
请不要在每行末尾输出额外的空格，否则您的解决方案可能会被视为错误！

示例

标准输入	标准输出
2	2
3	0 1 1
4	2 2 1
	2 1 2
	4
	1 1 2 2
	1 2 3 2
	4 1 4 3
	4 4 3 3

备注

测试用例示例如下。



## 问题 1. 计数器

输入文件：标准输入  
输出文件：标准输出  
时间限制 1 秒 内存限制  
1024 兆字节

计数器上有两个按钮。按下 "+" 按钮，计数器上的数值将增加 1

计数器的初始值为 0。

有人在计数器上进行了  $n$  次操作。每次操作都是按下两个按钮中的一个。已知有  $m$  个条件，其中第  $i$  个条件可以用两个整数  $a_i$  和  $b_i$  来描述，表示在  $a_i$ -th 操作后，计数器上的值是  $b_i$ 。

有没有办法按下按钮，使所有已知条件都得到满足？

### 输入

有多个测试用例。输入的第一行包含一个整数  $T$ ，表示测试用例的数量。对于每个测试用例

第一行包含两个整数  $n$  和  $m$  ( $1 \leq n \leq 10^9$ ,  $1 \leq m \leq 10^5$ )，表示操作数和已知条件数。

对于下面的  $m$  行，第  $i$  行包含两个整数  $a_i$  和  $b_i$  ( $1 \leq a_i \leq n$ ,  $0 \leq b_i \leq 10^9$ )，表示在  $a_i$ -th 运算之后，计数器上的值是  $b_i$ 。

保证所有测试用例的  $m$  之和不超过  $5 \times 10^5$ 。

### 输出

每个测试用例输出一行。如果有办法按下按钮，从而满足所有已知条件，则输出 "是"。否则输出 "否"。

### 示例

标准输入	标准输出
3 7 4 4 0 2 2 7 1 5 1 3 2 2 2 3 1 3 1 3 100	是 没有 没有

### 备注



对于第一个示例测试用例，按 "+cc+c+" 的顺序按键可以满足所有已知条件。对于第二个示例测试用例，有 8 种按 3 次按钮的方法。

印刷机	第 2 次行动结果	3-rd Op.	印刷机	第 2 次行动结果	3-rd Op.
ccc	0	0	+cc	0	0
cc+	0	1	+c+	0	1
c+c	1	0	++c	2	0
c++	1	2	+++	2	3

没有办法满足所有已知条件。

25-26 日

在第三个示例测试用例中，按 3 次按钮最多只能使计数器上的数值变为 3，而不可能是 100。



## 问题 J. 后缀结构

输入文件：	标准输入
输出文件：	标准输出
时间限制	2 秒 内存限制
	1024 兆字节

对于一个字符串  $u = u_1 \dots u_n$ ，让  $\text{pre}(u, i)$  成为前缀  $u_1 \dots u_i$ 。特别地， $\text{pre}(u, 0)$  是空字符串。

对于两个字符串  $u = u_1 \dots u_n$  和  $v = v_1 \dots v_m$ ，让  $u + v$  是连接符  $u_1 \dots u_n v_1 \dots v_m$ 。

给你一个长度为  $m$  的字符串  $t = t_1 \dots t_m$  和一棵树  $T$ ，树上有  $(n + 1)$  个顶点，分别用  $0, 1, \dots, n$  标示。根植于顶点  $0$ 。每条边都与一个字符相关联。请注意，在这个问题中，字母表可能包含超过 26 个字符。

考虑以下函数

$$f(i, j) = \max\{d(x) \mid s_x \text{ 是 } s \text{ 的后缀}_i + \text{pre}(t, j)\}$$

其中， $s_i$  是指从根节点到顶点  $i$  的最短路径上的字符串， $d(i)$  是指从根节点到顶点  $i$  的最短路径上的边数。

你的任务是计算  $g_1, g_2, \dots, g_m$  其中  $g_j = \sum_{i=1}^n f(i, j)$ 。

请注意， $s_0$  是空字符串，而空字符串是任何字符串的后缀。

### 输入

有多个测试用例。输入的第一行包含一个整数  $T$ ，表示测试用例的数量。对于每个测试用例

第一行包含两个整数  $n$  和  $m$  ( $1 \leq n, m \leq 2 \times 10^5$ )。

第二行包含  $n$  个整数  $p_1, p_2, \dots, p_n$  ( $0 \leq p_i < i$ )，其中  $p_i$  表示顶点  $i$  的父节点。第三行包含  $n$  个整数  $c_1, c_2, \dots, c_n$  ( $1 \leq c_i \leq n$ )，其中  $c_i$  表示来自顶点的边

顶点  $i$  的  $p_i$  与字母表中的第  $c_i$ -个字符相关联。保证  $p_i \neq p_j$  或  $c_i \neq c_j$ ，适用于所有  $i \neq j$ 。

第四行包含  $m$  个整数  $t_1, t_2, \dots, t_m$  ( $1 \leq t_i \leq n$ ) 其中  $t_i$  是字符串  $t$  的第  $i$  个字符。保证  $n$  和  $m$  的总和都不会超过  $2 \times 10^5$ 。

### 输出

对每个测试用例输出一行，其中包含  $m$  个整数  $g_1, g_2, \dots, g_m$ ，中间用空格隔开。



请不要在每行末尾输出额外的空格，否则您的解决方案可能会被视为错误！

示例

标准输入	标准输出
2	17 26 22
11 3	8 5 5 5 5 5 5 5 5 5 5 5 10 5
0 1 2 0 4 5 4 6 0 9 10	
1 3 2 2 1 3 4 1 3 2 1	
3 2 4	
5 16	
0 0 0 1 4	
1 2 3 2 2	
2 1 3 3 2 1 3 2 1 3 2 2 1 1 2 1	

让我们来计算第一个示例测试用例中的  $f(11, 1)$  和  $f(11, 2)$ ，以帮助您进一步理解。我们有  $s_{11} = \{3, 2, 1\}$ ，所以  $s_{11} + \text{pre}(t, 1) = \{3, 2, 1, 3\}$ 。由于  $s_6 = \{2, 1, 3\}$  是树中存在的最长后缀，因此  $f(11, 1) = d(6) = 3$ 。另外， $s_{11} + \text{pre}(t, 2) = \{3, 2, 1, 3, 2\}$  和  $s_3 = \{1, 3, 2\}$  是其在树中存在的最长后缀，所以  $f(11, 2) = d(3) = 3$ 。

## 问题 K. 总决赛

输入文件：标准输入  
输出文件：标准输出  
时间限制 1 秒 内存限制  
1024 兆字节

在 "猪国", "杀猪"是最受欢迎的roguelike游戏, 玩家使用卡组挑战名为 "邪恶土豆领主 "的老大。  
一般游戏规则如下：

1. 游戏开始时, 玩家手中有一组初始牌和一个从上到下排列的抽牌堆。
2. 在任何时候, 玩家的手牌和抽牌堆是唯一存在纸牌的地方。
3. 玩家可以打出手中的牌, 每打出一张牌都会导致这张牌先被丢弃, 然后触发某些效果。
4. 玩家只有在触发前一张牌的所有效果后, 才能使用下一张牌。

在这个问题中, 为了简单起见, 我们只考虑抽牌的效果。抽牌规则如下：

1. 在使用可以抽取其他牌的牌时, 玩家会从抽牌堆顶端按顺序抽取一定数量的牌到手中。
2. 玩家的手牌数量限制为  $k$ , 在任何时候, 玩家手牌的数量都不能超过  $k$  的限制。
3. 当玩家试图从抽牌堆顶端抽牌, 而手牌中已经有  $k$  张牌时, 抽出的牌将从抽牌堆中丢弃, 不会触发任何特效, 也不会加入手牌中。
4. 当玩家试图抽一张牌, 但抽牌堆是空的, 则什么也不会发生。

基于关键卡牌 "大结局"的卡组是游戏中最强大的卡组。一旦打出这张牌, 就会对所有敌人造成巨大伤害, 往往能取得游戏的胜利。不过, 要打出 "大结局 "卡牌有严格的条件。也就是说, 玩家的抽牌堆必须是空的, 这意味着所有的牌要么已经打出, 要么已经丢弃, 要么已经在玩家的手牌中。

Bie-Bot 是猪国仅次于神秘的奥斯卡的最聪明的猪, 他还会玩一套以总决赛为基础的卡组, 其中可以包含以下四种卡牌：



1. **大结局**：游戏中最强大的卡牌，打出这张牌就能轻松获胜，而 Bie-Bot 的卡组中只有一张**大结局**。这张牌只有在抽牌堆中没有牌时才能打出。

2. 快斩: 使用这张牌, 你可以从抽牌堆中抽取<sup>25,26 日</sup>一张牌。
3. 后空翻: 使用这张牌, 你可以从抽牌堆中抽取两张牌。
4. 伤痕: 状态牌, 一旦进入手牌就不能再使用。

游戏开始时, 小机器人很幸运地抽到了起手牌中唯一的一张 "压轴大牌", 而且他还知道抽牌堆中从上到下的每一张牌。现在, 他的目标是成功打出 "大结局" 这张牌。Bie-Bot 想知道, 在他的最优策略下, 实现目标所需的最小手牌大小限制  $k$  是多少。作为 Pigeland 第三聪明的玩家, 你能帮助他吗?

更正式地说, 您会得到一个长度为  $n$  的字符串  $S_H$ , 代表 Bie-Bot 的起手牌, 以及一个长度为  $m$  的字符串  $S_P$ , 代表从上到下的和牌堆。这两个字符串都由大写字母 "G"、"Q"、"B" 和 "W" 组成, 分别表示起手牌或和牌堆中相应位置的牌是 "大结局"、"快刀斩乱麻"、"后空翻" 或 "受伤"。Bie-Bot 可以根据上述规则使用手中的牌。请输出最小手牌大小限制  $k$  ( $k \geq n$ ), 以便 Bie-Bot 最终能打出大结局, 或说明不存在这样的  $k$ 。

### 输入

有多个测试用例。输入的第一行包含一个整数  $T$ , 表示测试用例的数量。对于每个测试用例

第一行包含两个整数  $n$  和  $m$  ( $1 \leq n, m \leq 2500$ ), 表示 Bie-Bot 起始手牌和抽牌堆中的牌数。

第二行包含长度为  $n$  的字符串  $S_H$ , 代表 Bie-Bot 的起手牌, 由大写字母 "G"、"Q"、"B" 和 "W" 组成。保证  $S_H$  中正好有一个 "G"。

输入的第三行包含一个长度为  $m$  的字符串  $S_P$ , 代表 Bie-Bot 从上到下的抽纸堆, 由大写字母 "Q"、"B" 和 "W" 组成。

保证所有测试用例的  $(n + m)$  之和不大于  $5 \times 10^4$ 。

### 输出

对每个测试用例输出一行, 其中包含一个整数, 表示能成功演奏 *Grand Finale* 的最小手牌大小  $k$  ( $k \geq n$ ), 如果不能演奏 *Grand Finale*, 则为 IMPOSSIBLE。

### 示例

标准输入	标准输出
2 2 6 BG BQWBWW 4 6 GQBW	3 不可能



备注

我们用 "手/牌 "字符串来表示这种情况。对于第一个样本测试案例，一种最优策略是：

- 1.  $BG/BQWBW \xrightarrow{B} BQG/WBWW$
- 2.  $BQG/WBWW \xrightarrow{Q} BWG/BWW$
- 3.  $BWG/BWW \xrightarrow{B} BWG/W$ （由于手牌大小限制，此处放弃一个 "W"）。

4.  $BWG/W \xrightarrow{B} WWG/\emptyset$  (此处只能抽一张牌，因为抽牌堆中已经没有牌了。)
5.  $WWG/\emptyset \xrightarrow{G}$  成功进行大结局!





# 问题 L. 电梯

输入文件：	标准输入
输出文件：	标准输出
时间限制	1 秒 内存限制
	1024 兆字节

有  $n$  组包裹等待投递。第  $i$  组有  $c_i$  个包裹，每个包裹的重量相同，为  $w_i$  ( $w_i$  为 1 或 2)，应送到第  $f_i$  - 层。

有一部电梯，每次可运送总重量不超过  $k$  ( $k$  为偶数) 的包裹。电梯从底层开始，逐渐向最高层  $h$  移动，最后回到底层。

更具体地说，设  $(w, f)$  是一个重量为  $w$  的包裹，应送到第  $f$  层。一个多集合 (允许重复元素的集合) 的包裹  $\mathbf{P}$  可以在同一车程内交付，如果

这趟旅程最多花费  $\sum_{(w,f) \in \mathbf{P}} f$  单位电力。

运送所有包裹至少需要多少电力？

请注意，每次搭乘可包含不同组别的包裹，每组包裹都可交付通过多次乘坐。你可以把这个问题当作总共有  $\sum_{i=1}^n c_i$  包裹要送达、

只是有些包裹的重量和目的地可能相同。

## 输入

有多个测试用例。输入的第一行包含一个整数  $T$ ，表示测试用例的数量。对于每个测试用例

第一行包含两个整数  $n$  和  $k$  ( $1 \leq n \leq 10^5$ ， $2 \leq k \leq 2 \times 10^{10}$ ， $k$  为偶数)，表示电梯的组数和容量。

对于下面的  $n$  行，第  $i$  行包含三个整数  $c_i$ 、 $w_i$  和  $f_i$  ( $1 \leq c_i \leq 10^5$ ， $w_i \in \{1, 2\}$ ， $1 \leq f_i \leq 10^5$ )，表示第  $i$  组包裹的数量、每个包裹的重量和包裹的目的地。

保证所有测试用例的  $n$  之和不超过  $3 \times 10^5$ 。

## 输出

对每个测试用例输出一行，其中包含一个整数，表示运送所有包裹所需的最小总电力单位。

标准输入	标准输出
2	24
4 6	100000
1 1 8	
7 2 5	
1 1 7	
3 2 6	
8 1200000	
100000 1 100000	
100000 1 12345	
100000 2 100000	
100000 2 12345	
100000 1 100000	
100000 1 12345	
100000 2 100000	
100000 2 12345	

备注

对于第一个示例测试用例，我们可以采用这种策略：

- 在第一次骑行中，运送包裹（2，6）、（2，6）和（2，5）。这次运送需要消耗 6 个单位的电能。
- 在第二次骑行中，运送包裹（1，8）、（1，7）、（2，6）和（2，5）。这次运送需要消耗 8 个单位的电能。
- 在第三次骑行中，运送包裹（2，5）、（2，5）和（2，5）。这次运送需要消耗 5 个单位的电能。
- 在第四次骑行中，运送包裹（2，5）和（2，5）。这次运送需要消耗 5 个单位的电能。

总电力单位为  $6 + 8 + 5 + 5 = 24$ 。可以证明，这就是所需的最小总电力单位。

在第二个测试案例中，所有包裹都可以在同一车程内送达。

## 问题 M. 收集雨水

输入文件：标准输入

输出文件：标准输出

时间限制 5 秒 内存限制

1024 兆字节

有一个直方图，由长度为  $n$  的整数序列  $a_1, a_2, \dots, a_n$  表示。对于从左到右的第  $i$  个条形图，其高度为  $a_i$ ，宽度为 1。

我们将对直方图进行  $q$  次修改。第  $i$  次修改可以用一对整数  $(x_i, v_i)$  表示，我们将把第  $x_i$  - 条的高度增加  $v_i$ 。

每次修改后，请回答以下问题：计算如果一场大雨倾盆而下，并尽可能填满所有凹坑，这个柱状图能截留多少水。

更具体地说，给定长度为  $n$  的整数序列  $a_1, a_2, \dots, a_n$ ，第  $i$  次修改后， $a_{x_i}$  将增加  $v_i$ 。每次修改后，请回答以下问题：设  $f_i = \max(a_1, a_2, \dots, a_i)$  和  $g_i = \max(a_i, a_{i+1}, \dots, a_n)$ ，计算

$$\sum_{i=1}^n (\min(f_i, g_i) - a_i)$$

## 输入

有多个测试用例。输入的第一行包含一个整数  $T$ ，表示测试用例的数量。对于每个测试用例

第一行包含一个整数  $n$  ( $1 \leq n \leq 10^5$ )，表示直方图中的条数。

第二行包含  $n$  个整数  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^6$ )，其中  $a_i$  表示第  $i$  条栏的初始高度。

第三行包含一个整数  $q$  ( $1 \leq q \leq 10^5$ )，表示修改的次数。

对于下面的  $q$  行，第  $i$  行包含两个整数  $x_i$  和  $v_i$  ( $1 \leq x_i \leq n, 1 \leq v_i \leq 10^6$ )，表示第  $i$  次修改使第  $x_i$  -th 条的高度增加了  $v_i$ 。

保证所有测试用例的  $n$  和  $q$  之和都不超过  $10^6$ 。

## 输出

对每个修改输出一行，其中包含一个整数，表示该直方图可以截留多少雨水。

## 示例

标准输入	标准输出
2	1



6子	4
1 2 3 4 5 6	25, 26 日
2	180
1 2	
3 3	
5	
100 10 1 10    100	
1	
3 100	