

“科大国创杯”

2025 年安徽省青少年信息学科普日活动

初中组试题

时间：2025 年 4 月 19 日 14:00 ~ 18:00

题目名称	足球联赛	果汁	旅行	抽卡
题目类型	传统型	传统型	传统型	传统型
目录	match	juice	trip	card
可执行文件名	match	juice	trip	card
输入文件名	match.in	juice.in	trip.in	card.in
输出文件名	match.out	juice.out	trip.out	card.out
测试点时限	1.0 秒	1.0 秒	1.0 秒	1.0 秒
内存限制	512 MiB	512 MiB	512 MiB	512 MiB
数据组数	10	10	20	25
对于 C++	match.cpp	juice.cpp	trip.cpp	card.cpp
对于 C++	-O2 -std=c++14			

注意事项与提醒：

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. 函数 main() 的返回值类型必须是 int，程序正常结束时的返回值必须是 0。
3. 选手需要在桌面上建立以选手的参赛号为名的目录，并由选手为每道题再单独建立一个子目录，子目录名与对应的试题英文名相同。选手提交的每道试题的源程序必须存放在相应的子目录下。
4. 因违反以上三点而出现的错误或问题，申诉时一律不予受理。如果现场有不同要求，以现场为准。
5. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
6. 程序可使用的栈内存空间限制与题目的内存限制一致。
7. 只提供 Linux 格式附加样例文件。
8. 评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以其为准。

足球联赛 (match)

题目描述

小可可正在研究 W 市的足球联赛。

W 市的足球联赛有 n 支球队，球队编号分别为 $1, 2, 3 \dots n$ ，每个赛季，球队之间会进行比赛。

已知本赛季进行了 m 场比赛，第 i 场是 a_i, b_i 两支球队进行比赛，并且比分是 $c_i : d_i$ ，根据比分判定胜负，从而影响积分。

1. 如果 $c_i > d_i$ ，则 a_i 球队获胜。
2. 如果 $c_i < d_i$ ，则 b_i 球队获胜。
3. 如果 $c_i = d_i$ ，则 a_i 和 b_i 球队平局。

赛后积分变化：每场比赛如果分出胜负，则胜者得 3 分，败者不得分；否则平局双方各得 1 分。

需要求出 m 场比赛后每支球队的积分。

输入格式

第一行读入两个正整数 n, m 。

接下来 m 行，每行输入四个整数 a_i, b_i, c_i, d_i 。

输出格式

输出一行，第 i 个整数表示 i 球队积分。

样例 1 输入

```
4 5
1 2 7 0
2 3 1 2
4 2 3 1
1 3 3 0
3 4 1 1
```

样例 1 输出

```
6 0 4 4
```

样例 2/3/4/5 输入

详见选手文件夹下的 `match/match*.in` 文件。

样例 2/3/4/5 输出

详见选手文件夹下的 `match/match*.ans` 文件。

数据规模与约定

对于 20% 的数据，保证 $n = 2, m = 1$ 。

对于另外 30% 的数据，保证 $c_i \neq d_i$ ，不存在平局。

对于另外 20% 的数据，保证 $c_i = d_i$ ，只存在平局。

对于 100% 的数据，保证 $2 \leq n \leq 20, 1 \leq m \leq \frac{n(n-1)}{2}, 1 \leq a_i, b_i \leq n, a_i \neq b_i, 0 \leq c_i, d_i \leq 10$ ，任意两支球队最多进行一场比赛。

果汁 (juice)

题目描述

小可可买了美味的果汁，但他太懒了，于是要求奶龙给他倒果汁。

有 m 个水杯，第 i 个水杯容量为 V_i 。现在有 n 个单位体积的果汁，小可可要求奶龙一共倒 n 次（每次倒一个单位体积，正好倒完），第 j 次倒果汁只能倒进第 a_j 或 $a_j + 1$ 个水杯。具体的，如果第 a_j 和 $a_j + 1$ 两个水杯都不是满的，那么可以任意倒入其中一个；如果有且仅有一个不是满的，那么只能倒入不是满的那一个；如果都已经被倒满了，那么小奶龙就可以开心地喝掉这一个单位体积的果汁。并且为了方便，小可可会安排小奶龙从左向右倒果汁，也就是说对于所有 $1 \leq j < n$ ，保证 $a_j \leq a_{j+1}$ 。

小奶龙想知道自己最多能喝掉单位体积的果汁，你能帮帮他吗？

你一共需要解决 T 组测试数据。

输入格式

第一行一个正整数 T ，表示数据组数。

对于每组数据，第一行两个正整数 n, m ，分别表示一共倒 n 次以及有 m 个水杯。第二行 m 个正整数 V_i 。第三行 n 个正整数 a_i ，保证 a_i 单调不降。

输出格式

对于每组数据，输出一行一个数，表示小奶龙最多能喝掉多少单位体积的果汁。

样例 1 输入

```
1
3 3
1 1 1
1 2 2
```

样例 1 输出

```
1
```

样例 2 输入

详见选手文件夹下的 juice/juice*.in 文件。

样例 2 输出

详见选手文件夹下的 juice/juice*.ans 文件。

样例 1 解释

一共倒三次，第一次可以倒进第一个或者第二个水杯，后两次可以倒进第二个或者第三个水杯。第一次倒进第二个水杯，第二次倒进第三个，此时后两个水杯都满了，第三次没法倒，故答案为 1。显然没有更优的答案。

数据规模与约定

对于 20% 的数据，保证 $n, m \leq 2$ 。

对于 50% 的数据，保证 $n, m \leq 8$ 。

对于另外 20% 的数据，保证所有 $V_i = 1$ 。

对于另外 20% 的数据，保证所有 a_i 相同。

对于 100% 的数据，保证 $1 \leq T \leq 50, 1 \leq n \leq 3 \times 10^4, 2 \leq m \leq 3 \times 10^4, 1 \leq a_i < m, 1 \leq V_i \leq n$ 。且对于所有 $1 \leq i < n, a_i \leq a_{i+1}$ 。

旅行 (trip)

题目描述

小可可来到了 P 国旅行。

P 国共有 n 个城市和 m 条有向道路，其中第 i 条道路为从城市 u_i 到城市 v_i ，长度为 w_i 。**保证** $u_i < v_i$ 。

对于一次游览，假设小可可依次经过了城市 $x_1, x_2, \dots, x_k$ ，其中从 x_i 到 x_{i+1} 经过的路径长度为 y_i 。记 s_i 表示 $y_1, y_2, \dots, y_i$ 的**按位或**值。那么小可可认为这次游览就是从 x_1 走到 x_k ，疲劳度就是 $\text{mex}(s_1, s_2, \dots, s_{k-1})$ 。

其中 $\text{mex}(a_1, a_2, \dots, a_n)$ 表示最小的没有出现在 $a_1, a_2, \dots, a_n$ 中的自然数。例如 $\text{mex}(0, 2, 3) = 1$ ， $\text{mex}(1, 4) = 0$ ， $\text{mex}(0, 1, 2, 3, 4) = 5$ 。

小可可认为两座城市 s, t 之间的距离为他从 s 走到 t 所有可能的游览方案中疲劳度的最大值，记作 $d(s, t)$ 。如果不能从 s 走到 t ，那么 $d(s, t) = -1$ 。规定 $d(s, s) = 0$ 。现在他想要知道任意两座城市之间的距离之和，即 $\sum_{i=1}^n \sum_{j=1}^n d(i, j)$ 。

输入格式

第一行两个整数 n, m 。

接下来 m 行，每行三个整数，代表 u_i, v_i, w_i 。

输出格式

一行一个整数表示答案。

样例 1 输入

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

样例 1 输出

```
0
```

样例 2/3/4/5 输入

详见选手文件夹下的 trip/trip*.in 文件。

样例 2/3/4/5 输出

详见选手文件夹下的 trip/trip*.ans 文件。

样例 1 解释

当从 1 走到 3 时有两条路径。其中从 1 直接到 3 疲劳度为 $\text{mex}(2) = 0$ 。从 1 到 2 再到 3 疲劳度为 $\text{mex}(0, 1) = 2$ 。所以 1 到 3 的距离为 2。

以此类推，有 $d(1, 1) = d(2, 2) = d(3, 3) = 0$ 。 $d(2, 1) = d(3, 1) = d(3, 2) = -1$ 。 $d(1, 2) = 1, d(1, 3) = 2, d(2, 3) = 0$ 。总和为 0。

数据规模与约定

对于 10% 的数据，保证 $n \leq 3, m \leq 5$ 。

对于 25% 的数据，保证 $n \leq 20, m \leq 40$ 。

对于 45% 的数据，保证 $n \leq 300, m \leq 500$ 。

对于 60% 的数据，保证 $n \leq 3000, m \leq 5000$ 。

对于另外 10% 的数据，保证 $w_i \geq 1$ 。

对于另外 10% 的数据，保证 $m = n - 1, u_i = i$ 。

对于 100% 的数据，保证 $1 \leq n \leq 3 \times 10^4, 1 \leq m \leq 2 \times 10^5, 1 \leq u_i < v_i \leq n, 0 \leq w_i \leq 10^9$ 。

抽卡 (card)

题目描述

小可可正在和波特玩卡牌游戏。

这个游戏有 m 种卡牌。种类为 x 的卡牌价值也为 x 。首先，波特一共会进行 n 次抽卡来确定游戏的牌堆。第 i 次，会有 a_i 种新的卡牌解锁。也就是说如果第 i 次抽卡之前第 $1 \sim s$ 种卡牌解锁了，那么第 i 次时第 $1 \sim s + a_i$ 种卡牌都会解锁。保证 $\sum_{i=1}^n a_i = m$ 。初始没有卡牌解锁。

在第 i 次抽卡时，波特会等概率随机选择一种已经被解锁的卡牌，并且取出两张这个种类的卡牌，依次放在现在所有牌的右边。在 n 次抽卡结束后，一共会有 $2n$ 张卡牌从左向右排成一排作为游戏的牌堆。小可可知道这 $2n$ 张卡牌各自的价值。

现在，小可可和波特会轮流从这个牌堆中进行抽卡，直到抽完所有卡牌，小可可先手。波特每次抽卡只会取走当前牌堆中最左边的卡牌。而小可可可以在当前牌堆中任意选择一张取走。小可可希望抽卡结束后自己手中的卡牌价值总和最大。他想要知道，如果自己采取最优策略，那么期望得到的价值是多少呢？你只需要告诉他答案对 $10^9 + 7$ 取模的结果就行。

关于期望：设离散型随机变量 X 的概率分布为 $p_i = P\{X = x_i\}$ ，那么我们称 $E = \sum x_i p_i$ 的值为 X 的期望。不过在本题中，由于每种可能的情况出现概率相等，所以你可以简单地理解为所有方案中小可可得到的卡牌价值总和除以总方案数。

关于有理数取余：不难发现答案一定是一个有理数，设其为 $C = \frac{A}{B}$ ，其中 A, B 互质，你需要输出 $C \bmod 1000000007$ 的值。这个值被定义为 $Bx \equiv A \pmod{1000000007}$ 的最小非负整数解。

输入格式

第一行两个整数 n, m 。

第二行 n 个整数表示 a 。

输出格式

输出一行一个整数表示答案。

样例 1 输入

```
2 3
1 2
```


样例 1 输出

4

样例 2/3/4/5/6/7/8/9 输入

详见选手文件夹下的 card/card*.in 文件。

样例 2/3/4/5/6/7/8/9 输出

详见选手文件夹下的 card/card*.ans 文件。

样例 1 解释

第 1 次，解锁的卡牌种类只有 1，于是波特会取出两张 1 卡牌。

第 2 次，解锁的卡牌种类有 1, 2, 3。波特会随机取出某一种类的两张卡牌。于是有如下三种可能：

牌堆中的卡牌从左到右分别为：1, 1, 1, 1，概率为 $\frac{1}{3}$ ，小可可可以获得的最大价值为 $1 + 1 = 2$ 。

牌堆中的卡牌从左到右分别为：1, 1, 2, 2，概率为 $\frac{1}{3}$ ，小可可可以获得的最大价值为 $2 + 2 = 4$ 。

牌堆中的卡牌从左到右分别为：1, 1, 3, 3，概率为 $\frac{1}{3}$ ，小可可可以获得的最大价值为 $3 + 3 = 6$ 。

于是答案为 $2 \times \frac{1}{3} + 4 \times \frac{1}{3} + 6 \times \frac{1}{3} = 4$ 。

数据规模与约定

测试点编号	$n \leq$	$m \leq$	特殊性质
1 ~ 2	3	3	无
3 ~ 6	8	8	无
7 ~ 9	500	500	$a_1 = m$
10 ~ 13	500	500	无
14 ~ 15	500	10^5	无
16 ~ 19	100	10^9	无
20 ~ 21	500	10^9	$a_1 = m$
22 ~ 25	500	10^9	无

对于 100% 的数据，满足 $1 \leq n \leq 500, 1 \leq m \leq 10^9, 0 \leq a_i \leq m$ 。保证 $\sum_{i=1}^n a_i = m$ 并且 $a_1 > 0$ 。