

# ZJOI省选模拟

2398886573

题目名称	三塔合一	刺客信条	胖
源程序文件名	tower.pas/c/cpp	assassin.pas/c/cpp	joke.pas/c/cpp
输入文件名	tower.in	assassin.in	joke.in
输出文件名	tower.out	assassin.out	joke.out
每个测试点时限	1s	1s	3s
内存限制	128MB	256MB	512MB
测试点数目	7	10	10
是否有部分分	否	否	否
题目类型	传统型	传统型	传统型
是否有附加文件	否	否	否
判断方式	文本比较	文本比较	实数比较

注意：

1. 评测时的栈空间大小不做单独限制, 但使用的总空间大小不能超过内存限制。
2. 考试之中不要交流, AK 之后不要大声喧哗, 没有AK 的同学请安静做题。
3. 请大家完成试题, 不要倦生, 不要高喊“这题太难我不会”之类的话。
4. 本套题难度较大, 码量稍大, 覆盖知识点广, 题目又相对切合实际的背景, 解法相对比较自然。出题人相信, 这套美妙的题目, 可以为拼搏于OI之路的你, 提供一个有力的援助。

# 1 三塔合一

Input file: tower.in

Output file: tower.out

Time limit: 1 seconds

Memory limit: 128 megabytes

## 1.1 Tasks

小H是一个热爱矩阵的女孩子。

小H有一个金字塔，它的基座是一个边长为 $N$ 的正方形,如果把金字塔的高度描述成一个俯视图下的 $N \times N$  的整数矩阵,那么每个格子的高度值就等于它到正方形边缘的距离。

1	1	1	1	1
1	2	2	2	1
1	2	3	2	1
1	2	2	2	1
1	1	1	1	1

小H有一个螺旋塔，它的基座也是一个边长为 $N$ 的正方形,如果把它的高度描述成一个俯视图下的 $N \times N$  的整数矩阵,那么每个格子的高度值可以由以下方法确定:从左上角开始顺时针沿着边沿向内依次填入 $1, 2, 3, \dots, N^2$ 。

1	2	3	4	5
16	17	18	19	6
15	24	25	20	7
14	23	22	21	8
13	12	11	10	9

小H有一个反螺旋塔，它是有螺旋塔按相反的顺序填入的。

1	16	15	14	13
2	17	24	23	12
3	18	25	22	11
4	19	20	21	10
5	6	7	8	9

小H想把同大小的3种塔放在一起,把对应位置的3个数值乘起来形成一个新的矩阵。现在的问题是给一个矩形区域,问新矩阵中这块区域的数字的和是多少。

## 1.2 Description

### 1.2.1 Input

第一行正整数 $N, x_1, y_1, x_2, y_2$ 代表矩阵的大小和询问的矩阵的左上角和右下角的坐标。注意其中左上角的坐标为 $(1, 1)$ ，第一个维度代表行号，第二个代表列号。

### 1.2.2 Output

输出区域内三个塔的高度的积的和，对十进制26281314取模。

## 1.3 Sample

### 1.3.1 Input1

5 1 1 2 2

### 1.3.2 Output1

643

### 1.3.3 Input2

666 233 111 455 632

### 1.3.4 Output2

16240995

## 1.4 Constraint

对于15%的数据， $1 \leq N \leq 500$ 。

对于另外15%的数据， $x_1 = x_2, y_1 = y_2$ 。

对于另外15%的数据， $x_1 = y_1 = 1, x_2 = y_2 = N$ 。

对于100%的数据， $1 \leq N \leq 10^6$ 。

## 2 刺客信条

Input file: assassin.in

Output file: assassin.out

Time limit: 1 seconds

Memory limit: 256 megabytes

### 2.1 Tasks

小H是一个热爱历史的女孩子。这一天，她被一个历史故事吸引住了。

故事发生在1486年的意大利,Ezio原本只是一个文艺复兴时期的贵族,后来因为家族成员受到圣殿骑士的杀害,决心成为一名刺客。最终,凭借着他的努力和出众的天赋,成为了杰出的刺客大师,他不仅是身手敏捷的武林高手,飞檐走壁擅长各种暗杀术。刺客组织在他的带领下,为被剥削的平民声张正义,赶跑了原本统治意大利的圣殿骑士首领一教皇亚历山大六世。在他的一生中,经历了无数次惊心动魄、扣人心弦的探险和刺杀。

曾经有一次,为了寻找Altair留下的线索和装备,Ezio在佛罗伦萨中的刺客墓穴探索。这个刺客墓穴中有许多密室,且任何两个密室之间只存在一条唯一的路径。这些密室里都有一个刺客标记,他可以启动或者关闭该刺客标记。为了打开储存着线索和装备的储藏室,Ezio必须操作刺客标记来揭开古老的封印。要想解开这个封印,他需要通过改变某些刺客标记的启动情况,使得所有刺客标记与封印密码“看起来一样”。在这里,“看起来一样的定义是:存在一种“标记”密室与“密码”密室之间一一对应的关系,使得密室间的连接情况和启动情况相同(提示中有更详细解释)。幸运的是,在Ezio来到刺客墓穴之前,在Da Vinci的帮助下,Ezio已经得知了打开储藏室所需要的密码。

而你的任务则是帮助Ezio找出达成目标所需要最少的改动标记次数。

小H想让你来解决这个历史问题,以便满足她的好奇心。

### 2.2 Description

#### 2.2.1 Input

第一行给出一个整数 $n$ ,表示密室的个数。

第二行至第 $n$ 行,每行给出两个整数 $a$ 和 $b$ ,表示第 $a$ 个密室和第 $b$ 个密室之间存在一条通道。

第 $n+1$ 行,给出 $n$ 个整数,分别表示当时每个密室的启动情况(0表示关闭,1表示启动)。

第 $n + 2$  行，给出 $n$  个整数，分别表示密码中每个密室的启动情况。

### 2.2.2 Output

输出只有一行，即输出最少改动标记次数。

## 2.3 Sample

### 2.3.1 Input1

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

### 2.3.2 Output1

```
1
```

### 2.3.3 Input2

```
7
1 2
1 3
2 4
2 5
3 6
3 7
1 1 1 0 0 1 1
1 0 1 1 1 0 0
```

### 2.3.4 Output2

```
1
```

## 2.4 Explanation

密室的编号是可以变的！在样例一中将第三个密室关闭后，在当前标记和密码之

间, 存在 $1 \rightarrow 4, 2 \rightarrow 3, 3 \rightarrow 2, 4 \rightarrow 1$  的对应关系, 重新编号后连接情况没有改变, 且标记与密码对应。对于更一般的情况, 存在一个1 到 $n$  的置换 $P$ , 使得对于任意密室之间的道路( $u \rightarrow v$ ), 都一定有密码密室中的道路( $P(u) \rightarrow P(v)$ ); 如果不存在密室之间的道路( $u \rightarrow v$ ), 则一定没有密码密室中的道路( $P(u) \rightarrow P(v)$ )。

## 2.5 Constraint

对于30%的数据,  $n \leq 10$ 。

对于60%的数据,  $n \leq 100$ 。

对于100%的数据,  $n \leq 700$ , 且每个密室至多与11 个密室相通。

## 3 胖

Input file: joke.in

Output file: joke.out

Time limit: 3 seconds

Memory limit: 512 megabytes

### 3.1 Tasks

小H是九条可怜的好朋友，也是这题的主人公。

小H是一个富有的女孩子。她住在著名的The Place（宫殿）中。

小H是一个努力的女孩子。她每天都做着不一样的题来锻炼她的The Salt（灵魂）。

这天，她打算在她的宫殿外围修筑一道城墙，城墙上会有 $n$ 座瞭望塔。你可以把城墙看做一条线段，瞭望塔是线段上的 $n$ 个点，其中1和 $n$ 分别为城墙的两个端点。其中第 $i$ 座瞭望塔和第 $i + 1$ 座瞭望塔的距离为 $w_i$ ，它们之间的道路是双向的。

城墙很快就修建好了，现在小H开始计划修筑他的宫殿到城墙的道路。因为这题的题目名称，小H打算用她的宫殿到每一个瞭望塔的最短道路之和来衡量一个修建计划。

现在小H手上有 $m$ 个设计方案，第 $k$ 个设计方案会在宫殿和瞭望塔之间修建 $T_k$ 条双向道路，第 $i$ 条道路连接着瞭望塔 $a_i$ ，长度为 $l_i$ 。

计算到每一个瞭望塔的最短路之和是一个繁重的工程，本来小H想用广为流传的SPFA算法来求解，但是因为她的buffer（缓冲区）实在是太小了，她只能转用原始的贝尔福特曼算法来计算，算法的流程大概如下：

1. 定义宫殿是0号点，第 $i$ 个瞭望塔是 $i$ 号点，双向边 $u_i, v_i, l_i$ 为一条连接 $u_i$ 和 $v_i$ 的双向道路。令 $d$ 为距离数组，最开始 $d_0 = 0, d_i = 10^{18} (i \in [1, n])$

2. 令辅助数组 $c = d$ 。依次对于每一条边 $u_i, v_i, w_i$ 进行增广， $c_{u_i} = \min(c_{u_i}, d_{v_i} + w_i), c_{v_i} = \min(c_{v_i}, d_{u_i} + w_i)$ 。

3. 令 $t$ 为 $c$ 和 $d$ 中不一样位置个数，即令 $S = \{i | c_i \neq d_i\}$ ，则 $t = |S|$ 。若 $t = 0$ 说明 $d$ 就是最终的最短路，算法结束。否则令 $d = c$ ，回到第二步。

因为需要计算的设计方案实在是太多了，所以小H雇佣了一些人来帮他进行计算。为了避免这些人用捏造出来的数据偷懒，她定义一个设计方案的校验值为在这个方案上运行贝尔福特曼算法每一次进入第三步 $t$ 的和。她会让好几个雇佣来的人计算同样的设计方案，并比对每一个人给出的校验值。

你是小H雇佣来的苦力之一，聪明的你发现这个情形下计算最短路的长度的和是一件非常简单的事情。但是寄人篱下不得不低头，你不得不再计算出每一个方案

的校验值来交差。

## 3.2 Description

### 3.2.1 Input

第一行输入两个整数 $n, m$ ，表示瞭望塔个数和设计方案个数。

接下来一行 $n - 1$ 个数 $w_i$ ，表示瞭望塔 $i$ 和 $i + 1$ 之间道路的长度。

接下来 $m$ 行，每行描述一个设计方案。第一个整数 $K$ 表示设计方案中的道路数量，接下来 $K$ 个数对 $a_i, l_i$ 为一条宫殿到瞭望塔的边。

### 3.2.2 Output

对于每一个设计方案，输出一行一个整数表示校验值。

## 3.3 Sample

### 3.3.1 Input

```
5 5
2 3 1 4
1 2 2
2 1 1 4 10
3 1 1 3 1 5 1
3 1 10 2 100 5 1
5 1 1 2 1 3 1 4 1 5 1
```

### 3.3.2 Output

```
5
8
5
8
5
```

## 3.4 Explanation

对于第一个设计方案，每一个阶段 $d$ 的变化为：



$$\begin{aligned}
& -[0, 10^{18}, 10^{18}, 10^{18}, 10^{18}, 10^{18}] \rightarrow [0, 10^{18}, 2, 10^{18}, 10^{18}, 10^{18}] \rightarrow [0, 4, 2, 5, 10^{18}, 10^{18}] \\
& \rightarrow [0, 4, 2, 5, 6, 10^{18}] \rightarrow [0, 4, 2, 5, 6, 10]
\end{aligned}$$

因此校验值为  $1 + 2 + 1 + 1 = 5$

对于第二个设计方案，每一个阶段  $d$  的变化为：

$$\begin{aligned}
& -[0, 10^{18}, 10^{18}, 10^{18}, 10^{18}, 10^{18}] \rightarrow [0, 1, 10^{18}, 10^{18}, 10^{18}, 10^{18}] \\
& \rightarrow [0, 1, 3, 11, 10, 14] \rightarrow [0, 1, 3, 6, 10, 14] \rightarrow [0, 1, 3, 6, 7, 11]
\end{aligned}$$

因此校验值为  $2 + 3 + 1 + 1 + 1 = 8$

对于第三个设计方案，每一个阶段  $d$  的变化为：

$$-[0, 10^{18}, 10^{18}, 10^{18}, 10^{18}, 10^{18}] \rightarrow [0, 1, 10^{18}, 1, 10^{18}, 1] \rightarrow [0, 1, 3, 1, 2, 1]$$

因此校验值为  $3 + 1 + 1 = 5$

对于第四个设计方案，每一个阶段  $d$  的变化为：

$$\begin{aligned}
& -[0, 10^{18}, 10^{18}, 10^{18}, 10^{18}, 10^{18}] \rightarrow [0, 10, 100, 10^{18}, 10^{18}, 1] \rightarrow [0, 10, 12, 103, 5, 1] \\
& \rightarrow [0, 10, 12, 6, 5, 1] \rightarrow [0, 10, 9, 6, 5, 1]
\end{aligned}$$

因此校验值为  $3 + 3 + 1 + 1 = 8$

对于第五个设计方案，每一个阶段  $d$  的变化为：

$$-[0, 10^{18}, 10^{18}, 10^{18}, 10^{18}, 10^{18}] \rightarrow [0, 1, 1, 1, 1, 1]$$

因此校验值为 5

### 3.5 Constraint

测试点	$n$	$m$	$K$	其他约定
1	$\leq 1000$	$\leq 1000$	$\leq 100$	无
2	$\leq 1000$	$\leq 1000$	$\leq 100$	无
3	$\leq 2 \times 10^5$	$\leq 2 \times 10^5$	$\leq 100$	无
4	$\leq 2 \times 10^5$	$\leq 2 \times 10^5$	$\leq 100$	无
5	$\leq 2 \times 10^5$	$\leq 2 \times 10^5$	$\leq 2 \times 10^5$	$1 \leq w_i, l_i \leq 50$
6	$\leq 2 \times 10^5$	$\leq 2 \times 10^5$	$\leq 2 \times 10^5$	$1 \leq w_i, l_i \leq 50$
7	$\leq 2 \times 10^5$	$\leq 2 \times 10^5$	$\leq 2 \times 10^5$	无
8	$\leq 2 \times 10^5$	$\leq 2 \times 10^5$	$\leq 2 \times 10^5$	无
9	$\leq 2 \times 10^5$	$\leq 2 \times 10^5$	$\leq 2 \times 10^5$	无
10	$\leq 2 \times 10^5$	$\leq 2 \times 10^5$	$\leq 2 \times 10^5$	无

对于100%的数据，保证每个设计方案 $1 \leq a_i \leq n$ 。

对于100%的数据，保证 $1 \leq w_i, l_i \leq 10^9, 1 \leq \sum K \leq 2 \times 10^5$ 。