

重庆育才信息学奥林匹克测试试题

提高组

(请选手务必仔细阅读本页内容)

一. 题目概况

中文题目名称	瑞士轮	对称二叉树	关押罪犯	道路游戏
英文题目名	swiss	tree	prison	game
可执行文件名	swiss	tree	prison	game
输入文件名	swiss.in	tree.in	prison.in	game.in
输出文件名	swiss.out	tree.out	prison.out	game.out
每个测试点时限	0.2 秒	0.5 秒	0.2 秒	0.2 秒
测试点数目	10	25	10	20
每个测试点分值	10	4	10	5
附加样例文件	有	有	有	有
结果比较方式	全文比较 (过滤行末空格及文末回车)			
题目类型	传统	传统	传统	传统
内存上限	128M	256MB	128MB	128MB

注意事项:

- 1、文件名 (程序名和输入输出文件名) 必须使用英文小写。
- 2、C/C++中函数 `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、特别提醒: 评测在 NOI Linux 下进行。

1. 瑞士轮

(`swiss.cpp/c/pas`)

【背景】

在双人对决的竞技性比赛, 如乒乓球、羽毛球、国际象棋中, 最常见的赛制是淘汰赛和循环赛。前者的特点是比赛场数少, 每场都紧张刺激, 但偶然性较高。后者的特点是较为公平, 偶然性较低, 但比赛过程往往十分冗长。

本题中介绍的瑞士轮赛制, 因最早使用于 1895 年在瑞士举办的国际象棋比赛而得名。它可以看作是淘汰赛与循环赛的折衷, 既保证了比赛的稳定性, 又能使赛程不至于过长。

【问题描述】

$2*N$ 名编号为 $1\sim 2N$ 的选手共进行 R 轮比赛。每轮比赛开始前, 以及所有比赛结束后, 都会按照总分从高到低对选手进行一次排名。选手的总分为第一轮开始前的初始分数加上已参加过的所有比赛的得分和。总分相同的, 约定编号较小的选手排名靠前。

每轮比赛的对阵安排与该轮比赛开始前的排名有关：第 1 名和第 2 名、第 3 名和第 4 名、……、第 $2K-1$ 名和第 $2K$ 名、……、第 $2N-1$ 名和第 $2N$ 名，各进行一场比赛。每场比赛胜者得 1 分，负者得 0 分。也就是说除了首轮以外，其它轮比赛的安排均不能事先确定，而是要取决于选手在之前比赛中的表现。

现给定每个选手的初始分数及其实力值，试计算在 R 轮比赛过后，排名第 Q 的选手编号是多少。我们假设选手的实力值两两不同，且每场比赛中实力值较高的总能获胜。

【输入】

输入文件名为 `swiss.in`。

输入的第一行是三个正整数 N 、 R 、 Q ，每两个数之间用一个空格隔开，表示有 $2*N$ 名选手、 R 轮比赛，以及我们关心的名次 Q 。

第二行是 $2*N$ 个非负整数 s_1, s_2, \dots, s_{2N} ，每两个数之间用一个空格隔开，其中 s_i 表示编号为 i 的选手的初始分数。

第三行是 $2*N$ 个正整数 w_1, w_2, \dots, w_{2N} ，每两个数之间用一个空格隔开，其中 w_i 表示编号为 i 的选手的实力值。

【输出】

输出文件名为 `swiss.out`。

输出只有一行，包含一个整数，即 R 轮比赛结束后，排名第 Q 的选手的编号。

【输入输出样例】

<code>swiss.in</code>	<code>swiss.out</code>
2 4 2 7 6 6 7 10 5 20 15	1

【输入输出样例说明】

	本轮对阵	本轮结束后的得分			
选手编号	/	①	②	③	④
初始	/	7	6	6	7
第 1 轮	①—④ ②—③	7	6	7	8
第 2 轮	④—① ③—②	7	6	8	9
第 3 轮	④—③ ①—②	8	6	9	9
第 4 轮	③—④ ①—②	9	6	10	9

【数据范围】

对于 30% 的数据， $1 \leq N \leq 100$;

对于 50% 的数据， $1 \leq N \leq 10,000$;

对于 100% 的数据， $1 \leq N \leq 100,000$ ， $1 \leq R \leq 50$ ， $1 \leq Q \leq 2N$ ， $0 \leq s_1, s_2, \dots, s_{2N} \leq 10^8$ ， $1 \leq w_1, w_2, \dots, w_{2N} \leq 10^8$ 。

2. 对称二叉树

(`tree.cpp/c/pas`)

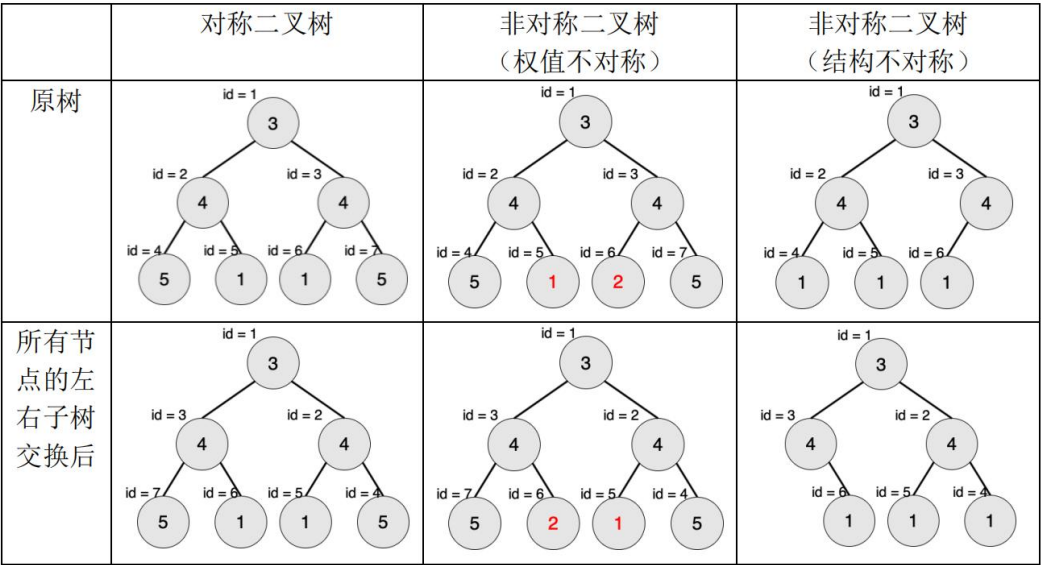
【问题描述】

一棵有点权的有根树如果满足以下条件，则被轩轩称为**对称二叉树**：

1. 二叉树；

2. 将这棵树**所有**节点的左右子树交换，新树和原树对应位置的结构相同且点权相等。

下图中节点内的数字为权值，节点外的 id 表示节点编号。



现在给出一棵二叉树，希望你找出它的一棵子树，**该子树为对称二叉树，且节点数最多**。请输出这棵子树的节点数。

注意：只有树根的树也是对称二叉树。本题中约定，以节点 T 为子树根的一棵“子树”指的是：节点 T 和它的**全部**后代节点构成的二叉树。

【输入格式】

输入文件名为 tree.in。

第一行一个正整数 n，表示给定的树的节点的数目，规定节点编号 1~n，其中节点 1 是树根。

第二行 n 个正整数，用一个空格分隔，第 i 个正整数 v_i 代表节点 i 的权值。

接下来 n 行，每行两个正整数 l_i, T_i ，分别表示节点 i 的左右孩子的编号。如果不存在左/右孩子，则以 -1 表示。两个数之间用一个空格隔开。

【输出格式】

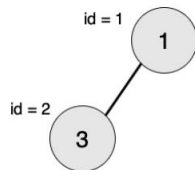
输出文件名为 tree.out。

输出文件共一行，包含一个整数，表示给定的树的最大对称二叉子树的节点数。

【输入输出样例 1】

tree in	tree out
2 1 3 2 -1 -1 -1	1

【输入输出样例 1 说明】

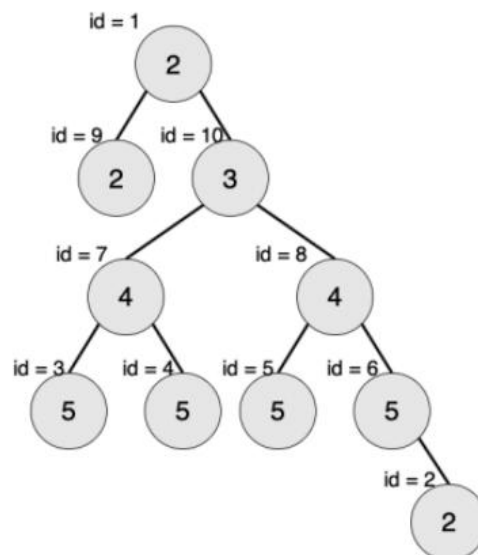


最大的对称二叉子树为以节点 2 为树根的子树，节点数为 1。

【输入输出样例 2】

tree in	tree out
10	3
2 2 5 5 5 5 4 4 2 3	
9 10	
-1 -1	
-1 -1	
-1 -1	
-1 -1	
-1 2	
3 4	
5 6	
-1 -1	
7 8	

【输入输出样例 2 说明】



最大的对称二叉子树为以节点 7 为树根的子树，节点数为 3。

【数据规模与约定】

共 25 个测试点。

$v_i \leq 1000$ 。

测试点 1~3, $n \leq 10$, 保证根结点的左子树的所有节点都没有右孩子, 根结点的右子树的所有节点都没有左孩子。

测试点 4~8, $n \leq 10$ 。

测试点 9~12, $n \leq 105$, 保证输入是一棵“满二叉树”。

测试点 13~16, $n \leq 105$, 保证输入是一棵“完全二叉树”。

测试点 17~20, $n \leq 105$, 保证输入的树的点权均为 1。

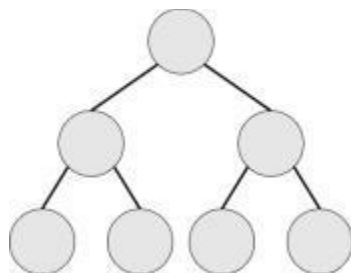
测试点 21~25, $n \leq 106$ 。

本题约定:

层次: 节点的层次从根开始定义起, 根为第一层, 根的孩子为第二层。树中任一节点的层次等于其父亲节点的层次加 1。

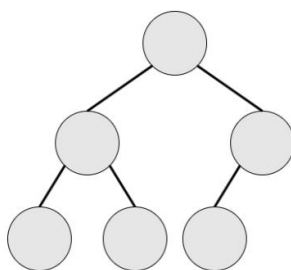
树的深度： 树中节点的最大层次称为树的深度。

满二叉树： 设二叉树的深度为 h ，且二叉树有 $2^h - 1$ 个节点，这就是满二叉树。

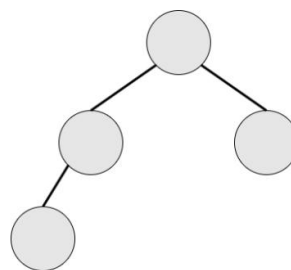


满二叉树（深度为 3）

完全二叉树： 设二叉树的深度为 h ，除第 h 层外，其它各层的结点数都达到最大 个数，第 h 层所有的结点都连续集中在最左边，这就是完全二叉树。



完全二叉树（深度为 3）



完全二叉树（深度为 3）

3. 关押罪犯

(prison.pas/c/cpp)

【问题描述】

S 城现有两座监狱， 一共关押着 N 名罪犯， 编号分别为 $1 \sim N$ 。他们之间的关系自然也极不和谐。很多罪犯之间甚至积怨已久， 如果客观条件具备则随时可能爆发冲突。我们用“怨气值”（一个正整数值）来表示某两名罪犯之间的仇恨程度，怨气值越大，则这两名罪犯之间的积怨越多。如果两名怨气值为 c 的罪犯被关押在同一监狱， 他们俩之间会发生摩擦， 并造成影响力为 c 的冲突事件。

每年年末， 警察局会将本年内监狱中的所有冲突事件按影响力从大到小排成一个列表， 然后上报到 S 城 Z 市长那里。公务繁忙的 Z 市长只会去看列表中的第一个事件的影响力， 如果影响很坏， 他就会考虑撤换警察局长。

在详细考察了 N 名罪犯间的矛盾关系后， 警察局长觉得压力巨大。他准备将罪犯们在两座监狱内重新分配， 以求产生的冲突事件影响力都较小， 从而保住自己的乌纱帽。假设只要处于同一监狱内的某两个罪犯间有仇恨， 那么他们一定会在每年的某个时候发生摩擦。那么， 应如何分配罪犯， 才能使 Z 市长看到的那个冲突事件的影响力最小？ 这个最小值是多少？

【输入】

输入文件名为 `prison.in`。输入文件的每行中两个数之间用一个空格隔开。

第一行为两个正整数 N 和 M ，分别表示罪犯的数目以及存在仇恨的罪犯对数。

接下来的 M 行每行为三个正整数 a_j, b_j, c_j ，表示 a_j 号和 b_j 号罪犯之间存在仇恨，其怨气值为 c_j 。数据保证 $1 \leq a_j < b_j \leq N$ ， $0 < c_j \leq 1,000,000,000$ ，且每对罪犯组合只出现一次。

【输出】

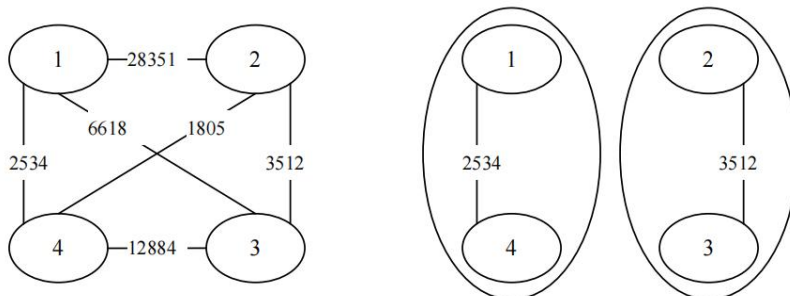
输出文件 `prison.out` 共 1 行，为 Z 市长看到的那个冲突事件的影响力。如果本年内监狱中未发生任何冲突事件，请输出 0。

【输入输出样例】

<code>prison.in</code>	<code>prison.out</code>
4 6 1 4 2534 2 3 3512 1 2 28351 1 3 6618 2 4 1805 3 4 12884	3512

【输入输出样例说明】

罪犯之间的怨气值如下面左图所示，右图所示为罪犯的分配方法，市长看到的冲突事件影响力是 3512（由 2 号和 3 号罪犯引发）。其他任何分法都不会比这个分法更优。

**【数据范围】**

对于 30% 的数据有 $N \leq 15$ 。

对于 70% 的数据有 $N \leq 2000$ ， $M \leq 50000$ 。

对于 100% 的数据有 $N \leq 20000$ ， $M \leq 100000$ 。

4. 道路游戏

(game.pas/c/cpp)

【问题描述】

小新正在玩一个简单的电脑游戏。

游戏中有一条环形马路，马路上有 n 个机器人工厂，两个相邻机器人工厂之间由一小段马路连接。小新以某个机器人工厂为起点，按顺时针顺序依次将这 n 个机器人工厂编号为 $1 \sim n$ ，因为马路是环形的，所以第 n 个机器人工厂和第 1 个机器人工厂是由一段马路连接在一起的。小新将连接机器人工厂的这 n 段马路也编号为 $1 \sim n$ ，并规定第 i 段马路连接第 i 个机器人工厂和第 $i+1$ 个机器人工厂 ($1 \leq i \leq n-1$)，第 n 段马路连接第 n 个机器人工厂和第 1 个机器人工厂。

游戏过程中，每个单位时间内，每段马路上都会出现一些金币，金币的数量会随着时间发生变化，即不同单位时间内同一段马路上出现的金币数量可能是不同的。小新需要机器人的帮助才能收集到马路上的金币。所需的机器人必须在机器人工厂用一些金币来购买，机器人一旦被购买，便会沿着环形马路按顺时针方向一直行走，在每个单位时间内行走一次，即从当前所在的机器人工厂到达相邻的下一个机器人工厂，并将经过的马路上的所有金币收集给小新，例如，小新在 i ($1 \leq i \leq n$) 号机器人工厂购买了一个机器人，这个机器人会从 i 号机器人工厂开始，顺时针在马路上行走，第一次行走会经过 i 号马路，到达 $i+1$ 号机器人工厂 (如果 $i=n$ ，机器人会到达第 1 个机器人工厂)，并将 i 号马路上的所有金币收集给小新。

游戏中，环形马路上不能同时存在 2 个或者 2 个以上的机器人，并且每个机器人最多能够在环形马路上行走 p 次。小新购买机器人的同时，需要给这个机器人设定行走次数，行走次数可以为 $1 \sim p$ 之间的任意整数。当马路上的机器人行走完规定的次数之后会自动消失，小新**必须立刻**在任意一个机器人工厂中购买一个新的机器人，并给新的机器人设定新的行走次数。

以下是游戏的一些补充说明：

1. 游戏从小新第一次购买机器人开始计时。
2. 购买机器人和设定机器人的行走次数是瞬间完成的，不需要花费时间。
3. 购买机器人和机器人行走是两个独立的过程，机器人行走时不能购买机器人，购买完机器人并且设定机器人行走次数之后机器人才能行走。
4. 在同一个机器人工厂购买机器人的花费是相同的，但是在不同机器人工厂购买机器人的花费不一定相同。
5. 购买机器人花费的金币，在游戏结束时再从小新收集的金币中扣除，所以在游戏过程中小新不用担心因金币不足，无法购买机器人而导致游戏无法进行。也因为如此，游戏结束后，收集的金币数量可能为负。

现在已知每段马路上每个单位时间内出现的金币数量和在每个机器人工厂购买机器人需要的花费，请你告诉小新，经过 m 个单位时间后，扣除购买机器人的花费，小新最多能收集到多少金币。

【输入】

输入文件名为 game.in。

第一行 3 个正整数， n ， m ， p ，意义如题目所述。

接下来的 n 行，每行有 m 个正整数，每两个整数之间用一个空格隔开，其中第 i 行描述了 i 号马路上每个单位时间内出现的金币数量 ($1 \leq \text{金币数量} \leq 100$)，即第 i 行的第 j ($1 \leq j \leq m$) 个数表示第 j 个单位时间内 i 号马路上出现的金币数量。

最后一行，有 n 个整数，每两个整数之间用一个空格隔开，其中第 i 个数表示在 i 号机器人工厂购买机器人需要花费的金币数量 ($1 \leq \text{金币数量} \leq 100$)。

【输出】

输出文件 `game.out` 共一行，包含 1 个整数，表示在 m 个单位时间内，扣除购买机器人花费的金币之后，小新最多能收集到多少金币。

【输入输出样例】

<code>game.in</code>	<code>game.out</code>
2 3 2 1 2 3 2 3 4 1 2	5

【数据范围】

对于 40% 的数据， $2 \leq n \leq 40$ ， $1 \leq m \leq 40$ 。

对于 90% 的数据， $2 \leq n \leq 200$ ， $1 \leq m \leq 200$ 。

对于 100% 的数据， $2 \leq n \leq 1000$ ， $1 \leq m \leq 1000$ ， $1 \leq p \leq m$ 。