浅谈随机化在信息学竞赛中的应用

——广东省韶关市第一中学 刘家骅

【摘要】

如今信息学竞赛题目日新月异，各种新型算法层出不穷，而作为一种新兴的算法，随机化算法在信息学竞赛这个舞台上也发挥着独特而重要的作用。

本文通过几个典型例题简单介绍随机化算法在信息学竞赛中的应用。

【关键字】

信息学竞赛 随机化

【引言】

伴随着信息学竞赛的发展，“随机化”一词在参加信息学竞赛的同学心中经历了一个由陌生到熟悉的过程。

随机化算法以其特有的灵活多变，逐渐在越来越多不同类型的竞赛题目中得到巧妙运用，在越来越多样化的信息学竞赛中拥有独有的优势。

掌握了随机化算法，无疑手上又多了一把解决问题的利刃。

简单题目的另类算法

例题：Geometrical dreams(Ural1046)

有一个多边形 $A_1A_2...A_N$ ，在每条边 A_iA_{i+1} 上向多边形外做一个等腰三角形 $A_iM_iA_{i+1}$ 使得角 $A_iM_iA_{i+1}=\alpha_i$。由 α_i 组成的集合满足其任何非空子集的角度和不是 360 度的倍数。给出 N，所有 M_i 的坐标和 α_i，写一个程序，输出多边行的顶点的坐标。

分析：

这道题目用简单的解方程就能够解决。

让我们从另外一个角度思考问题，从题目的条件易知，只要确定了一个顶点的坐标，多边形的其他顶点的坐标就能够通过简单计算得到，那么问题就转化为确定多边形的一个顶点的坐标。

如何确定一个顶点的坐标呢，枚举和二分等常用算法都无法为我们解决问题，于是，我们想到了随机化。

我们开始时将第一个点放在原点，通过计算能够得出第 N+1 个顶

点的坐标，如果第 N+1 个顶点和第 1 个顶点重合，这个多边形就是所求，但是显然这样的可能性非常渺茫，于是我们需要调整第一点的位置。显然，第一个点距离第 N+1 个点的距离越小，其位置越接近其实际位置。我们每次可以在暂时确定第一个点的位置附近随机一个点，判断第一个点放在这里这个位置时与原来相比第一个点与第 N+1 个点的距离是否比原来更小，如果是，则将第一个点的位置暂时定在这个位置，然后继续上述操作，直至第一个点与第 N+1 个点重合，我们就确定了多边形的顶点的坐标了。

事实证明，这种方法能够通过这道题目。虽然这样的方法显然在任何方面都要比前面提到的普通做法要复杂，对于解决这道题目没有太大的意义，但是，它提供给我们一种崭新的思路——随机化。

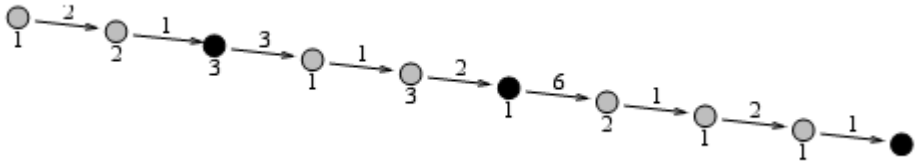
随机化算法对于这样的题目没有优势，但是，它在很多问题上都能得到运用，下面，我们一起来进一步领略随机化算法的魅力吧。

小试牛刀

例题：Two sawmills(CEOI2004)

从山顶到山脚的路上有 n 棵老树，现在政府决定砍掉它们，为了不浪费木材，每一棵树都会被转运到锯木场。

树只能往一个方向运输，向下。在路的尽头有一个锯木场。两个额外的锯木场可以在路上的任意一棵老树的位置上，你必须选择在哪里建造，使得运输的费用达到最少。运输费用是一分每米每千克木材。



分析：

这道题目的标准算法将数据转化为图象，用栈进行处理求出两个矩形的最大覆盖面积，时间复杂度为 O(N)。但是，这种算法对能力要求不小，不太容易想到。

我们看下随机化算法在这题上的表现。

首先最容易想到的随机化当然就是直接随机寻找两个点，计算出以这两个点为锯木场时的总运费，多次随机后将总费用最小的输出。

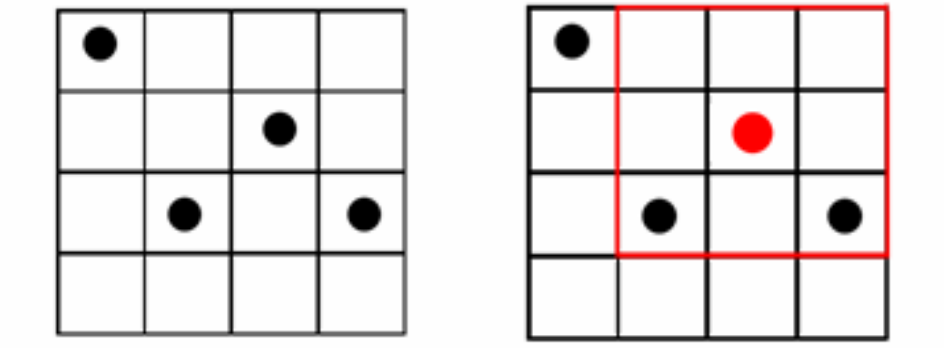
我们可以进行预处理，将计算的时间复杂度降为 O(1)，那么在时限内我们可以随机几百万次甚至几千万次，但是相对于总状态的四亿

来说，寻找到最优解的几率不是很大。

有没有更好的方法呢？

我们刚才是用随机化算法直接出解，准确性不太好，为了增加准确性，那么我们尝试一下用随机化来缩小区域范围。

我们建立一个矩阵 P，P[X,Y]表示第一个锯木场建立在 X，第二个锯木场建立在 Y 时的总运费。一开始时，矩阵的边长为 N。我们随机寻找一定数量的点(如下左图所示，取点数量应该充分利用时限并且注意效率，由于矩阵的大小一直在变化，推荐使用矩阵大小的定比确定取点数量)，计算出它们的值，取其最小点，以这个点为新矩阵的中心，以现在矩阵的边长的 3/4 的长度为新矩形的边长（如下右图所示），从原来的矩阵中取出一块作为新矩阵的范围(若新矩阵的范围出了原矩阵的边界就将其向里移动到原矩阵内)，然后继续在新矩阵中重复这样的操作，直至新矩阵足够小时，我们即可枚举新矩阵上的每一个点，取其中最小值作为答案。



我们惊喜地发现，这种随机化算法对于测试数据能够全部通过！

通过这道题目可以看出，随机化算法的灵活多变使得它的具有更为广阔的运用范围，在许多看似难以入手的地方通过巧妙地运用发光发热。而这样的多变性也使得我们需要灵活恰当地运用随机化算法才能发挥出它的优势。随机化算法并不只是简单地随便乱来，使用随机化算法的时候与其他算法一样值得细细斟酌，需要匠心独运。

下面，让我们来看看随机化算法在实际比赛中的运用解析。

实战解析

例题：小 H 的聚会(NOI2005)

小 H 从小就非常喜欢计算机，上了中学以后，他更是迷上了计算机编程。经过多年的不懈努力，小 H 幸运的被选入信息竞赛省队，就要

去他日思夜想的河南郑州参加第 22 届全国信息学奥林匹克竞赛（NOI 2005）。

小 H 的好朋友小 Y 和小 Z 得了这个消息，都由衷的为他感到高兴。他们准备举办一个 party，邀请小 H 和他的所有朋友参加，为小 H 庆祝一下。 经好几天的调查，小 Y 和小 Z 列出了一个小 H 所有好友的名单，上面一共有 N 个人（方便起见，我们将他们编号为 1 至 N 的整）。然而名单上的人实在是太多了，而且其中不少人小 Y 和小 Z 并不认识。如何把他们都组织起来参加聚会呢？

小 Y 和小 Z 希望为小 H 的 N 个好友设计一张联系的网络，这样，若某个人得知了关于聚会的最新情况，则其他人都可以直接或间接得到消息。同时为了尽量的保证消息传递得简单、高效以及最重要的一点：保密（为了给小 H 一个惊喜,在 party 的筹备阶段这个聚会的消息是绝对不能让他知道的），小 Y 和小 Z 决定让尽量少的好友直接联系：为了保证 N 个好友都能互能直接或间接联系到，只需要让(N-1)对好友直接联系就可以了。

显然，名单上的好友也不都互相认识，而即使是两个互相认识的人，他们之间的熟悉程度也是有区别的。因此小 Y 和小 Z 又根据调查的结果，列出了一个好友间的关系表，表中标明了哪些人是可以直接联系的，而对于每一对可以互相联系的好友，小 Y 和小 Z 又为他们标出了联系的愉快程度。如 3 和 4 的关系非常好，因此标记他们之间的联系愉快程度为 10；而 1 和 3 是一般的朋友，则他们的愉快程度要小一些。上面的图 1 表示一个 N=5 的联系表，其中点表示名单上的好友，边则表示两个好友可以直接联系，边上的数字即为他们联系的愉快程度。

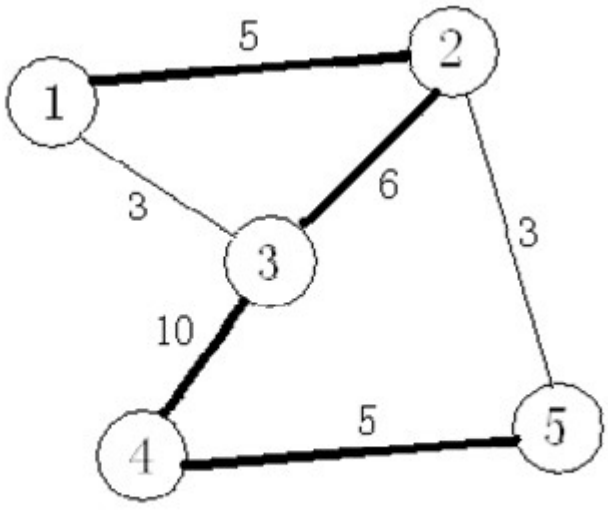


图 1

小 Y 和小 Z 希望大家都能喜欢这次聚会，因此决定在尽量最大化

联系网络的愉快程度：所谓联系网络的愉快程度，每一对直接联系人之间的愉快程度之和。如在图 1 中，加粗的边表示了一个让愉快程度最大联系的网络，其愉快程度为 5+6+10+5=26。

然而，如果让某个人直接和很多的人联系，这势必会给他增添很大的负担。因此小 Y 和小 Z 还为每个人分别设定了个最大的直接联系人数 ki，表示在联系网络中，最多只能有 k_i 个人和 i 直接联系。 还是用图 1 的例子，若我们为 1 至 5 每个点分别加上了 k_i = 1, 1, 4, 2, 2 的限制，则上述方案就不能满足要求了。此时的最优方案如图 2 所示，其愉快程度为 3+6+10+5 = 24。

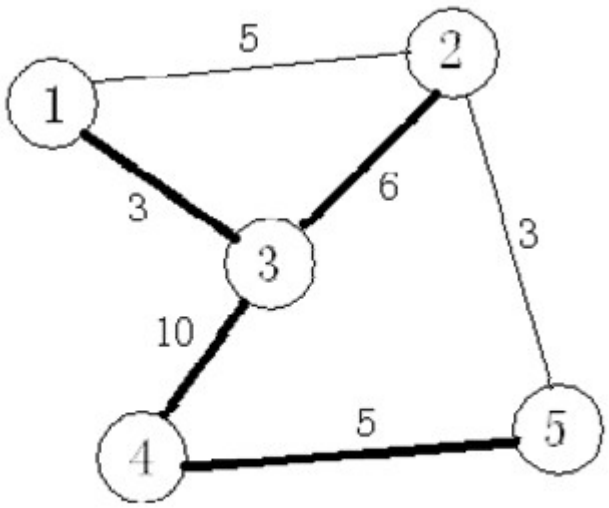


图 2

你能帮小 Y 和小 Z 求出在满足限制条件的前提下，愉快程度尽量大的一个联系网络吗？

分析：

这是一道提交答案题，题目数据分为三种类型，第一种类型包括第 1-3 个数据，每个点的度限制均为 N-1，等同于没有限制，直接用最小生成树算法即可解决。第二种类型包括第 4-6 个数据，其中 N-1 个点的度限制为 N-1，只有一个点有度限制，可以使用最小度限制生成树算法解决（《算法艺术与信息学竞赛》一书中有详细介绍）。

题目数据的第三种类型包括第 7-10 个数据，数据中所有点都有度限制，属于 NP 问题，而这个题目又是不要求最优解的开放性题目。这种问题实在是随机化算法自由翱翔的广阔天空。

我们可以采取随机化算法结合不同算法解决这个题目。

方法一：基于 Kruskal 的随机化算法

我们用 Kruskal 算法计算出符合点的度限制的一棵生成树，由于有

了度限制，所以这样得出的生成树不一定是最优的，我们可以在按边权大小排序后，再随机打乱部分边的顺序，再用 Kruskal 算法，以求得到更优的解。

在每次得到这样一棵较大的生成树后，我们再随机化选取一条不在生成树上的边，加入生成树上得到一个圈，再在这个圈上试图删除一条边使得生成树仍然满足度限制并且结果更优，如此重复多次，以得到不能再通过这种方法得到更优解的极优解。

多次使用 Kruskal 算法后输出计算出的最优解。

方法二：基于 Prim 的随机化算法

先用 Prim 算法计算出一个不考虑度限制的最大生成树。

在这棵生成树上，每次随机选取一个不满足度限制的点，并试图寻找一条不在生成树上边替代一条在生成树内并且连接到这个点的边以降低这个点的度。直至将这棵树修改到满足度限制或者无法再用这样的方法降低这棵树不满足度限制的点的度的时候停止。

重新在原最大生成树上重复上述操作多次，输出计算出的最优解。

两种算法的比较

		算法一		算法二	
数据	参考答案	稳定输出[1]	得分	稳定输出	得分
7	64034	62712	7	63856	9
8	570138	565529	8	563403	8
9	1021302	1021385	10	1021461	10
10	1041394	1041604	10	1042087	10

通过上面例题可以看出，使用随机化算法在开放性题目能够得到很好的结果，而使用随机化结合不同的算法会得到不同的结果，在解题时应运用恰当的算法配合随机化算法能以提高程序的准确性。

【总结】

从上述的题目中可以看出，随机化算法具有灵活多变的特点，应用随机化算法应该注意以下几点：

- 针对不同情况灵活运用随机化算法，充分发挥其灵活多变的特点
- 注意应用随机化算法与恰当的算法结合以激发更大的活力

[1] 稳定输出指多次运行时大多数情况得到的结果

- 注意随机决策和调整的效率，充分利用题目所给出的限制时间

随机化算法作为一种新兴的具有鲜明特点的算法，不仅能为我们提供解决很多题目的不同方法，在开放性题目上更具有明显的优势。灵活掌握和运用随机化算法无疑能够使我们在风云变幻的信息学赛场上更加应变自如。

【参考资料】

《算法艺术与信息学竞赛》 by 刘汝佳 黄亮
NOI2005 讲题大会幻灯片

【感谢】

感谢韶关市第一中学杨溢、孔德本、叶树雄同学的帮助
感谢林盛华老师的培养和魏皆老师的支持
感谢北极天南星对例题 1 的帮助

【附录】

原题：

Geometrical dreams

There is a polygon $A_1A_2\ldots A_N$ (the vertices A_i are numbered in clockwise order). On each side A_iA_{i+1} an isosceles triangle $A_iM_iA_{i+1}$ is built on the outer side of the polygon, and angle $A_iM_iA_{i+1} = \alpha_i$. Here we assume that $A_{N+1} = A_1$.

The set of angles α_i satisfies a condition that the sum of angles in any of its nonempty subsets is not aliquot to 360 degrees.

You are given N, coordinates of vertices Mi and angles α_i (measured in degrees). Write a program, which restores coordinates of the polygon vertices.

Input

The first line of the input contains an integer N ($3 \le N \le 50$). The next N lines contain pairs of real numbers x_i, y_i which are coordinates of points Mi ($-100 \le x_i, y_i \le 100$). And the last N lines of the input consist of degree values of angles α_i. All real numbers in the input contain at most 2 digits after decimal point.

Output

The output should contain N lines with points coordinates, i-th line should

contain the coordinates of A_i. Coordinates must be accurate to 2 digits after decimal point. You may assume that solution always exists.

Sample
input
3
0 2
3 3
2 0
90
90
90

output
1 1
1 3
3 1

Problem Author: Dmitry Filimonenkov
Problem Source: Ural State University collegiate programming contest (25.03.2000)

Two Sawmills

There are n old trees planted along a road that goes from the top of a hill to its bottom. Local government decided to cut them down. In order not to waste wood each tree should be transported to a sawmill.

Trees can be transported only in one direction: downwards. There is a sawmill at the lower end of the road. Two additional sawmills can be built along the road. You have to decide where to build them, as to minimize the cost of transportation. The transportation costs one cent per meter, per kilogram of wood.

Task
Write a program, that:

reads from the standard input the number of trees, their weights and locations,

calculates the minimum cost of transportation,

writes the result to the standard output.

Input
The first line of the input contains one integer n - the number of trees ($2 \le n \le 20\,000$). The trees are numbered 1,2,...,n starting from the top of the hill and going downwards. Each of the following n lines contains two positive integers separated by single space. Line i + 1 contains: w_i - weight (in kilograms) of the i-th tree and d_i - distance (in meters) between trees number i and i+1, $1 \le w_i \le 10\,000$, $0 \le d_i \le 10\,000$. The last of these numbers, d_n, is the distance from the tree number n to the lower end of the road. It is guaranteed that the total cost of transporting all trees to the sawmill at the end of the road is less than 2 000 000 000 cents.

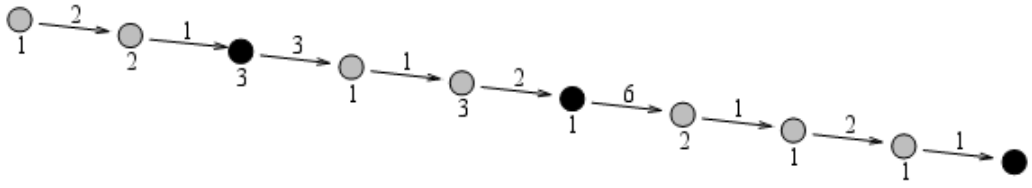
Output
The first and only line of output should contain one integer: the minimum cost of transportation.

Example
For the input data:
9
1 2
2 1
3 3
1 1
3 2
1 6
2 1
1 2
1 1

the correct result is:
26

The figure shows the optimal location of sawmills for the example data. Trees are depicted as circles with weights given below. Sawmills are marked black. The result is equal to:

$$1 \cdot (2+1) + 2 \cdot 1 + 1 \cdot (1+2) + 3 \cdot 2 + 2 \cdot (1+2+1) + 1 \cdot (2+1) + 1 \cdot 1$$



小 H 的聚会

任务描述

小 H 从小就非常喜欢计算机，上了中学以后，他更是迷上了计算机编程。经过多年的不懈努力，小 H 幸运的被选入信息竞赛省队，就要去他日思夜想的河南郑州参加第 22 届全国信息学奥林匹克竞赛（NOI2005）。

小 H 的好朋友小 Y 和小 Z 得了这个消息，都由衷的为他感到高兴。他们准备举办一个 party，邀请小 H 和他的所有朋友参加，为小 H 庆祝一下。 经好几天的调查，小 Y 和小 Z 列出了一个小 H 所有好友的名单，上面一共有 N 个人（方便起见，我们将他们编号为 1 至 N 的整）。然而名单上的人实在是太多了，而且其中不少人小 Y 和小 Z 并不认识。如何把他们都组织起来参加聚会呢？

小 Y 和小 Z 希望为小 H 的 N 个好友设计一张联系的网络，这样，若某个人得知了关于聚会的最新情况，则其他人都可以直接或间接得到消息。同时为了尽量的保证消息传递得简单、高效以及最重要的一点：保密（为了给小 H 一个惊喜,在 party 的筹备阶段这个聚会的消息是绝对不能让他知道的），小 Y 和小 Z 决定让尽量少的好友直接联系：为了保证 N 个好友都能互能直接或间接联系到，只需要让(N-1)对好友直接联系就可以了。

显然，名单上的好友也不都互相认识，而即使是两个互相认识的人，他们之间的熟悉程度也是有区别的。因此小 Y 和小 Z 又根据调查的结果，列出了一个好友间的关系表，表中标明了哪些人是可以直接联系的，而对于每一对可以互相联系的好友，小 Y 和小 Z 又为他们标出了联系的愉快程度。如 3 和 4 的关系非常好，因此标记他们之间的联系愉快程度为 10；而 1 和 3 是一般的朋友，则他们的愉快程度要小一些。上面的图 1 表示一个 N=5 的联系表，其中点表示名单上的好友，边则表示两个好友可以直接联系，边上的数字即为他们联系的愉快程度。

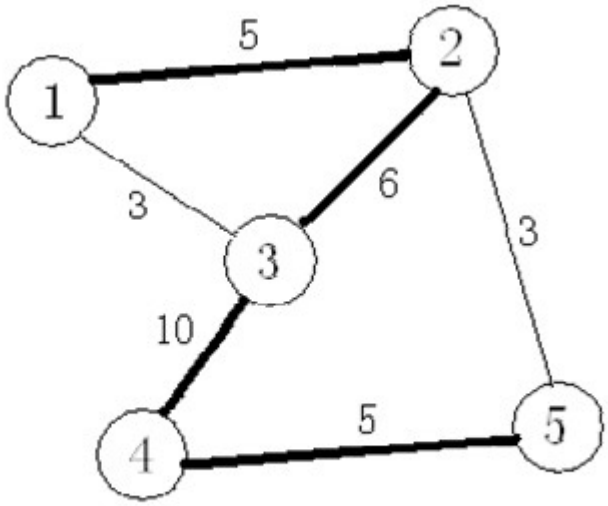


图 1

小 Y 和小 Z 希望大家都能喜欢这次聚会，因此决定在尽量最大化联系网络的愉快程度：所谓联系网络的愉快程度，每一对直接联系人之间的愉快程度之和。如在图 1 中，加粗的边表示了一个让愉快程度最大联系的网络，其愉快程度为 5+6+10+5=26。

然而，如果让某个人直接和很多的人联系，这势必会给他增添很大的负担。因此小 Y 和小 Z 还为每个人分别设定了个最大的直接联系人数 ki，表示在联系网络中，最多只能有 k_i 个人和 i 直接联系。 还是用图 1 的例子，若我们为 1 至 5 每个点分别加上了 k_i = 1, 1, 4, 2, 2 的限制，则上述方案就不能满足要求了。此时的最优方案如图 2 所示，其愉快程度为 3+6+10+5 = 24。

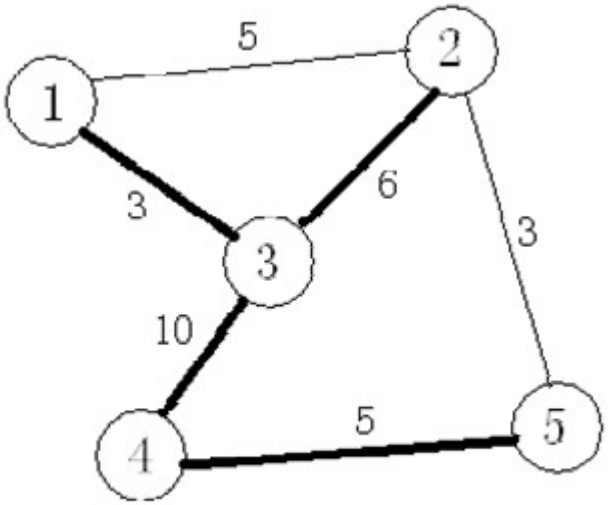


图 2

你能帮小 Y 和小 Z 求出在满足限制条件的前提下，愉快程度尽量大的一个联系网络吗？

输入格式

输入文件 party1.in 到 party10.in 已经放在用户目录中。

每个输入文件的第 1 行都是两个整数 N 和 M。N 表示小 H 的好友总数，M 表示小 Y 和小 Z 列出来的可以直接联系的好对数。

输入文件的第 2 行包含 N 个在[1, N-1]范围内的整数，依次描述 k1, k2, ..., kN。相邻的两个数字之间用一空格隔开。

以下 M 行，每行描述一对可以互相联系的好友，格式为 ui vi ci。表示 ui 和 vi 可以直接联系，他们的联系愉快程度为 ci。

另外，在所有这些数据的最后还有单独的一行包括一个(0, 1]范围内的实数 d 作为评分系数。你的程序并不需要去理会这个参数，但你可以根据这个参数的提示去设计不同的算法。有关 d 的说明，可以参见后面的评分方法。

输出格式

本题是一道提交答案式的题目，你需要提供十个输出文件从 party1.out 到 party10.out。

每个文件的第 1 行为一整数，表示你找到的最大的愉快程度。

以下(N-1)行，描述这个网络。每行一个数 e_i，表示在网络中，让输入文件中第(ei + 2)行描述的一对好友直接联系。

输入样例

```
5 6
1 1 4 2 2
1 2 5
1 3 3
2 3 6
2 5 3
3 4 10
4 5 5
0.00001
```

输出样例

```
24
2
3
5
6
```

样例说明

详见任务描述中的例子。

评分方法

本题设有部分分，对于每一个测试点：

如果你的输出方案不合法，即 ei 不符合范围或 ei 有重复或网络不连通等，该测试点得 0 分。

如果你输出的方案和输出文件第 1 行的愉快程度不一致，该测试点得 0 分。

否则该测试点得分按如下方法计算：设

$$a=(1-d)*our\_ans$$

$$b=(1+d*0.5)*our\_ans$$

如果你的结果小于 a，该测试点得 0 分；

如果你的结果大于 b，该测试点得 15 分；

否则你的得分为

$$your\_score=[(your\_ans-a)/(our\_ans)*10]$$

其中的 d 为评分系数（输入数据中最后一行的实数），our_ans 为我们提供的考解答，your_ans 为你的答案。

你如何测试自己的输出

我们提供 party_check 这个工具作为测试你的输出文件的办法。使用这个工具的方法是在控制台中输入：

./party_check <测试点编号 X>

在你调用这个程序后，party_check 将根据输入文件 partyX.in 和你的输出文件 partyX.out 给出测试的结果，其中包括：

Error: Not connected：你的程序输出的联系网络不连通；

Error: Edge xxx is duplicated：第 xxx 条边被输出了两次；

Error: Edge in Line xxx is out of range：你的程序在第 xxx 行输出的边的编号不在[1, M]范围内；

Error: Degree of Friend xxx is out of range：在联系网络中，和编号为 xxx 的好友直接联系的人超过了限　制；

Error: Scheme & happiness mismatch：方案和第一行的愉快程度不一致；

测试程序非法退出：其他情况；

Correct! Happiness = xxx：输出正确。

源程序：

例题 1 Geometrical dreams timus1046.pas
例题 2 Two Sawmills two.pas
例题 3 小 H 的聚会 方法一 party1.pas 方法二 party2.pas