



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 幸运的被选入信息学竞赛省队，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 得知了这个消息，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。

经过好几天的调查，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 的整数)。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 个好友都能互相直接或间接联系到，只需要让 $(N-1)$ 对好友直接联系就可以了。

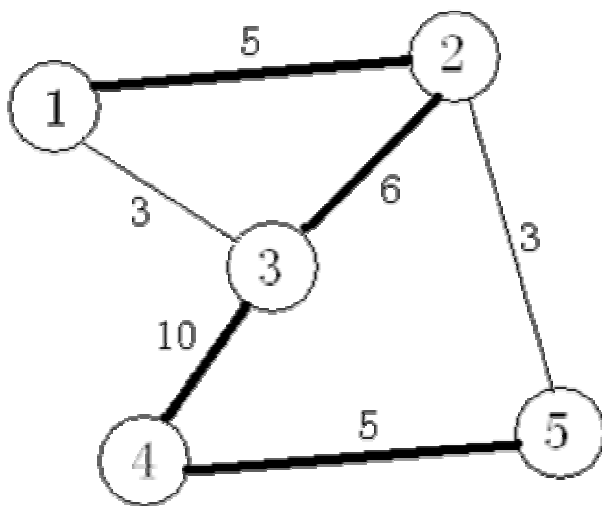


图 1

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。

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，即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一个最大的直接联系人数 k_i ，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 $3+6+10+5 = 24$ 。

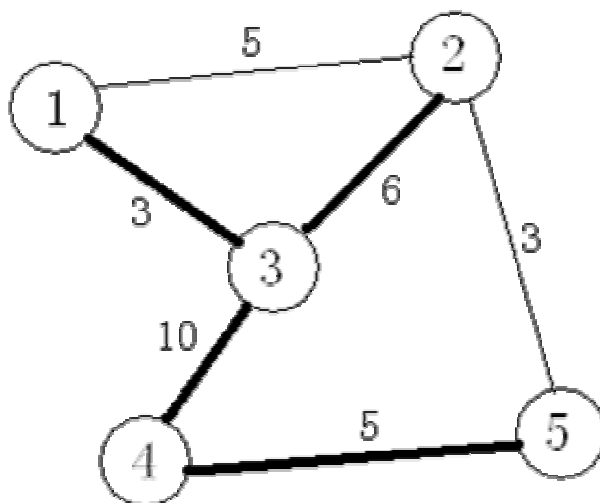


图 2

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友对数。

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 k_1, k_2, \dots, k_N 。相邻的两个数字之间用一个空格隔开。

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

另外，在所有这些数据之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 行为一个整数，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 $(e_i + 2)$ 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 e_i 不符合范围或 e_i 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

$$your_score = \left[\frac{your_ans - a}{our_ans - a} * 10 \right]$$

其中的 d 为评分系数（输入数据中最后一行的实数）， our_ans 为我们提供的参考解答， $your_ans$ 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。