

# NOI '05 P5 - Little H's Party

**Time limit:** 2.5s    **Memory limit:** 256M

## National Olympiad in Informatics, China, 2005

Little H has liked computers ever since he was little. Going to high school, he has become even more obsessed with computer programs. After many years of unremitting efforts, little H has fortunately been selected into the provincial informatics competition team. He is about to go to the National Olympiad in Informatics (NOI2005) held in Zhengzhou, Henan, which he has been dreaming about day and night.

Little H's good friends little Y and little Z has received the news, and both feel very happy for him. They prepare to hold a party, inviting little H and all of his friends to attend and celebrate.

After several days of research, little Y and little Z has come up with a list of all of little H's good friends. The list contains a total of  $N$  people (for simplicity, we label them with the integers from 1 to  $N$ ). However, there are truly too many people on the list, and among them, there are many whom little Y and little Z have never even met. Just how will they bring them all together to attend the party?

Little Y and little Z wish to set up a contact network for all  $N$  friends of little H. This way, if one person receives the latest information about the party, the others will all be able to directly or indirectly receive the news too. At the same time, these two must ensure that information is delivered with simplicity, efficiency, and most importantly: confidentiality (to give little H a surprise, news about the party must absolutely not reach him during the party's preparatory stages). Thus, little Y and little Z would like to minimize the number of direct contact links between the friends. **To guarantee that the  $N$  friends can directly or indirectly contact each other, only  $N - 1$  contact pairings are needed.**

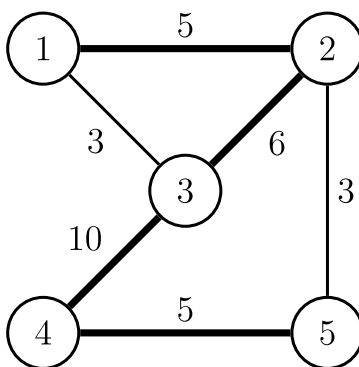


Figure 1

Clearly, not all of the friends on the list are mutually acquainted. Even for pairs of people that are acquainted, there are varying degrees of acquaintedness amongst different pairs. For this reason, little Y and little Z have used their research to generate a relationship chart between the friends, indicating which people can directly contact each other. Furthermore for each pair of friends in contact, little Y and little Z have also marked the **contact comfort level** between them. Since 3 and 4 share a really close friendship, the comfort level between them is recorded as 10. Since 1 and 3 are simply normal friends, the comfort level between them is smaller. Figure 1 above depicts a network for  $N = 5$ , where points represent people from the list of friends, lines indicate that two friends can directly contact, and numbers beside the lines specify the contact comfort level between them.

Little Y and little Z wish for everybody to enjoy this party, so they've decided to maximize the comfort level across the network: **the comfort level across the network refers to the sum of the comfort levels between pairs of**

**contactable friends.** For example in figure 1, the bold lines represent a network that maximizes the total comfort level, which is  $5 + 6 + 10 + 5 = 26$ .

However, if a person is given too many direct contacts, it is bound to be too heavy of a burden for them. Thus, little Y and little Z has assigned a number  $k_i$  for each person, indicating that there can be at most  $k_i$  people that can directly contact person  $i$  in the contact network.

Using the same example from figure 1, if we respectively assign person 1 to 5 the numbers  $k_i = 1, 1, 4, 2, 2$  as limits, then the method depicted above will not satisfy the requirements. At this point, the optimal method is the one depicted in figure 2, yielding a total comfort level of  $3 + 6 + 10 + 5 = 24$ .

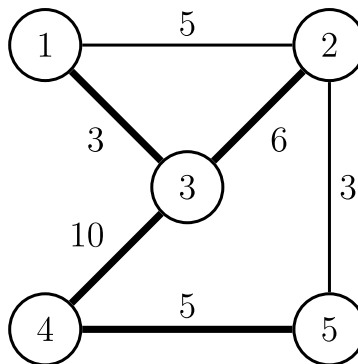


Figure 2

Can you help little Y and little Z determine the maximum total comfort level in the network, provided that all the above requirements are met?

## Input Specification

There are 10 test cases `party1.in` ~ `party10.in` that will be given to your program (through standard input). They can be downloaded here for you to study: [party.zip](#)

The first line of input contains the integer  $T$ , the test case number. For example, `party1.in` will have the single integer 1 as its first line. The program that you submit may choose to recognize this number (or ignore it), but must generate the output accordingly.

The second line of input contains two integers  $N$  and  $M$ .  $N$  is the total number of friends of little H, and  $M$  is the number of pairs of friends that can directly contact each other, as determined by little Y and little Z.

The third line of input contains  $N$  integers in the range  $[1, N - 1]$ , indicating the values of  $k_1, k_2, \dots, k_N$ . Adjacent integers are separated by a single space.

In the following  $M$  lines, each line describes a possible connection between a pair of friends in the format  $u_i v_i c_i$ . This indicates that  $u_i$  and  $v_i$  can directly contact each other, sharing a comfort level of  $c_i$ .

Also, the last line of the input contains a single real number  $d$  in the range  $(0, 1]$ , used as a scoring factor. Your program is not required to do anything with this number, but you may consider using it to suggest different methods of computation. Regarding the use of  $d$ , see the scoring section below.

## Output Specification

The first line of output should contain a single integer, indicating the largest possible comfort level.

The following  $N - 1$  lines should describe this network. Each line should contain one integer  $e_i$ , indicating that the connection described by the  $(e_i + 3)^{\text{th}}$  line of input should be part of the network.

## Sample Input

---

```
0
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
```

## Sample Output

---

```
24
2
3
5
6
```

## Scoring

---

For each test case, your score out of 10 will be determined as follows:

- If your output is invalid, i.e.  $e_i$  is not within range,  $e_i$  is repeated, the network is not connected, etc. Then your score for the test case will be 0.
- If your output is inconsistent with the total comfort level on the first line, then your score will be 0.
- Otherwise, your score will be determined as follows. Let:  
 $a = (1 - d) \times \text{our\_ans}$   
 $b = (1 + d \times 0.5) \times \text{our\_ans}$ 
  - If your answer is less than  $a$ , then your score will be 0.
  - If your answer is more than  $b$ , then your score will be 10.
  - Otherwise, your score will be:

$$\text{your\_score} = \frac{(\text{your\_ans} - a)}{(\text{our\_ans} - a)} \times 10$$

In the above,  $d$  is the grading factor (the real number at the end of the input), `our_ans` is the official answer that we will use for grading, and `your_ans` is the answer you'll have determined.

Problem translated to English by [Alex](#).