Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen (LEA)
Prof. Dr. Ernst W. Mayr
Christian Müller, Philipp Hoffmann, Chris Pinkau, Stefan Toman

Summer Term 2015
Problem Set 4
07.05.2015

Algorithms for Programming Contests

This problem set is due by

Thursday, 21.05.2015, 6:00 a.m.

Try to solve all the problems and submit them at

`http://judge.informatik.tu-muenchen.de/`

This week's problems are:

SS15N04A Chocolate **2**

SS15N04B Bank Robbery **4**

SS15N04C Christmas Presents **6**

SS15N04D Beer Pipes **8**

SS15N04E Football Champion **10**

The following amount of points will be awarded for solving the problems.

Problem	SS15N04A	SS15N04B	SS15N04C	SS15N04D	SS15N04E
Difficulty	easy	easy	medium	medium	hard
Points	4	4	6	6	8

If the judge does not accept your solution but you are sure you solved it correctly, use the "request clarification" option. In your request, include:

- the name of the problem (by selecting it in the subject field)
- a verbose description of your approach to solve the problem
- the time you submitted the solution we should judge

We will check your submission and award you half the points if there is only a minor flaw in your code.

If you have any questions please ask by using the judge's clarification form.

SS15N04A Chocolate

Author: Philipp Hoffmann

When thinking about chocolate, Lea usually gets happy. But not today! She has no more chocolate at home and all supermarkets near her home have "delivery problems". Of course she immediately calls the chocolate factory to find out what caused the problem. On the phone she is told that the lead organizer of chocolate transportation systems, Mr. D. I. Abetis is sick. Because no one else knows how to properly manage the conveyor belts that transport the chocolate from one production step to another, the whole production is on hold.

Lea has only one choice: She will go to the factory and solve the problem! At the factory, she is presented with the following situation:

There are multiple chocolate fountains from which warm chocolate springs in unlimited quantity. At these fountains, chocolate is cooled down and then transported to the wrapping station. This transport happens via a network of conveyor belts. Each conveyor belt has a maximum amount of chocolate it can transport per hour. Conveyor belts can end either at one of the fountains, at one of the wrapping stations, or at a distribution station. At a distribution station, all chocolate that arrives via an incoming conveyor belt has to be distributed onto outgoing conveyor belts.

The running direction of each conveyor belt can be switched, so a belt between a and b can transport either from a to b or the other way. For simplicity, assume that the wrapping stations have no limit to the amount of chocolate they can wrap.

Given the network of conveyor belts, Lea has to figure out how much chocolate can be wrapped per hour without violating the maximum load on the conveyor belts.

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with four integers n k m l, the number of chocolate fountains n, distribution stations k, wrapping stations m and conveyor belts l. l lines follow describing conveyor belts, each containing three integers v_i w_i c_i where v_i and w_i are locations and c_i is the capacity of the conveyor belt connecting v_i and w_i. The locations will be described by integers where 1 to n are chocolate fountains, $n+1$ to $n+k$ are distribution stations, $n+k+1$ to $n+k+m$ are wrapping stations.

Output

For each test case, output one line containing "Case #i: d" where i is its number, starting at 1, and d is the maximum amount of chocolate that can be wrapped per hour.

Each line of the output should end with a line break.

Constraints

- $1 \le t \le 20$
- $1 \le n, m \le 100$
- $1 \le k \le 500$
- $1 \le l \le 50000$
- $1 \le v_i, w_i \le m + n + k$ for all $1 \le i \le l$
- $1 \le c_i \le 20$ for all $1 \le i \le l$
- The graph of locations is connected.

Sample Data

Input

```
2
2 2 2 5
1 3 1
2 3 1
3 4 1
4 5 1
4 6 1

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

Output

```
Case #1: 1
Case #2: 4
```

SS15N04B Bank Robbery

Author: Christian Müller

You just got word that a bank robber is robbing the NBEAM (National Bank with Enormous Amounts of Money). The police of course wants to catch him, but they can not storm the building since he has taken hostages. Their plan is to catch him on his way out of the country.

Unfortunately, the police only has a very limited amount of personnel, but they want to be absolutely sure to catch that bank robber.

To catch him, they can build roadblocks on a few roads and check every person using those roads. Since some roads are large and some small, they all need different amounts of policemen to be blocked.

At dinner, Lea's father complains that he has been appointed "Head of Strategic Positioning" but he has no idea how to distribute the policemen. Can you tell them if they have enough policemen to cover every possible route out of the country?

Input

The first line of the input contains an integer t. t test cases follow.

Each test case begins with a line containing three integers l, the amount of available policemen, n, the amount of intersections and m, the amount of roads. m lines follow, each consisting of three integers i, j, k, specifying a road from intersection i to j with k being the amount of policemen it takes to construct a roadblock on it. All roads are useable in both directions. The robber always starts at intersection 1 and wants to get to the border (intersection n). Every test case ends with a blank line.

Output

For each test case, print a line containing "Case #i: t" with i being the number of the test case, starting at 1, and t being "yes" if the minimal amount of policemen it takes to construct at least one road block on every path from intersection 1 to n is smaller or equal to l, "no" otherwise.

Constraints

- $1 \leq t \leq 20$
- $1 \leq l \leq 10000$
- $2 \leq n \leq 400$
- $1 \leq m \leq 100000$

- $1 \leq i, j \leq n$
- $1 \leq k \leq 10000$
- There will always be at least one possible path from 1 to n.

Sample Data

Input

```
3
10 2 3
2 2 3
1 2 4
1 1 3

2 3 3
1 3 5
1 2 5
3 3 5

8 2 3
2 2 4
1 2 5
1 1 4
```

Output

```
Case #1: yes
Case #2: no
Case #3: yes
```

SS15N04C Christmas Presents

Author: Stefan Toman

Approximately 25% of the people who buy christmas presents begin to search for presents as late as one week before christmas. Lea has a different approach to this problem: She buys all the presents long before and uses December to decide which presents will be assigned to which of her friends. Lea has some constraints when she creates the assignment:

- All of Lea's friends should get a present.
- To be fair, each of her friends gets at most one present.
- None of the presents will be divided into multiple parts.
- Presents should only be assigned to a friend if he or she will like the present.

Luckily, Lea knows exactly which of her friends will like each present. She wants to prepare the algorithm to distribute the presents among her friends early before christmas, too. Can you help her writing it?

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with a line containing two integers n, the number of Lea's friends, and m, the number of presents Lea bought. The friends and presents are numbered from 1 to n or m, respectively. n lines follow describing the preferences of her friends: The i-th line contains a string with the numbers of the presents friend i likes. The numbers are comma-separated and may be given as sections where the first and last present are separated by a dash. For instance the string "1,10,3,5-8" represents presents 1, 3, 5, 6, 7, 8 and 10. Note that this string may also be empty.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is "yes" if there is an assignment satisfying all constraints or "no" if there is no such assignment. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 1000$

- $1 \leq m \leq 1000$
- No presents will be mentioned several times per line of input.
- Sections of presents will always be given with the smaller index first.

Sample Data

Input

```
2
3 3
1
1-2
2,3

2 2
1
1
```

Output

```
Case #1: yes
Case #2: no
```

SS15N04D Beer Pipes

Author: Chris Pinkau

After a stressful work day, Lea enjoys a nice cold beverage while sitting on her couch in front of the TV. Like most of the people from the region she comes from, she usually enjoys a beer on these occasions. And after having a few sips of the exquisite golden liquid, she contemplates the work that is put behind brewing such a masterpiece. Thus, she decides to visit the **BIER** (**B**rewery of **I**nternational **E**xcellence and **R**elevance), one of the many local breweries, on the next day to learn a bit more about the process behind her favourite beverage. Apart from all the usual brewery tour, she meets Mr. Barley Hops, the CEO of **BIER**. Recognising Lea, he says (in a heavy German accent) "Guten Tag my dear Fräulein Lea. I have heard about you and your problem solving skills, maybe you can help us? The workers installed new pipes for the Bier. They were so drunk, every pipe has a different shape und we don't know how much Bier we can pump into the pipes." Lea immediately sees the problem: the beer is poured into a pipe on one end of the brewery and exits at one valve at the other end in a great cauldron. In between there is a whole system of pipes that are connected in a seemingly chaotic fashion and are all shaped very differently. The question at stake is to come up with the highest amount of beer that can be put through the system so that the beer cauldron at the end is filled with as much beer as possible. Unfortunately, Lea is very busy right now, so she wants you to take a look at the problem. Make sure you can help Mr. Hops because he will grant you a lifetime supply of **BIER** beer if your solution is optimal!

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with two integers n and m, n being the number of valves $\{v_1, \ldots, v_n\}$ that connect the pipes, and m the number of pipes in the system. m lines follow where line i consists of three integers a_i, b_i and k_i, and a double x_i, denoting that there is a pipe that connects the valves v_{a_i} and v_{b_i} whose cross section has the shape of a regular polygon with k_i sides and side length x_i. If k_i is equal to 0, then the pipe is cylindrical with radius x_i.

The maximal amount of beer that can flow through a pipe is measured by the area of its cross section.

The first valve, where the beer enters the pipe system, is v_1, the exit of the pipe system, at the large beer cauldron, is v_n.

Beer in the pipes can flow in both directions.

Output

For each test case, output one line containing "Case #i: y" where i is its number, starting at 1, and y is either the maximal amount of beer that can be poured into v_1, or "impossible" if that amount is 0.

The precision for this problem is 8 decimals, i.e., the value of y should be correct up to 10^{-8}.

Constraints

- $1 \leq t \leq 20$
- $3 \leq n \leq 1000$
- $1 \leq m \leq 2000$
- $1 \leq a_i, b_i \leq n$ for all $1 \leq i \leq m$
- $3 \leq k_i \leq 20$ or $k_i = 0$ for all $1 \leq i \leq m$
- $0 < x_i \leq 100$ for all $1 \leq i \leq m$

Sample Data

Input

```
2
4 5
1 2 0 3.3
1 3 3 1.5
2 3 0 2.2
2 4 5 4.1
3 4 4 2.5

3 2
1 2 0 1.2
1 2 4 2
```

Output

```
Case #1: 35.17122510390053
Case #2: impossible
```

SS15N04E Football Champion

Author: Philipp Hoffmann

Lea likes to chat a lot with her colleagues and friends. A good topic to talk about is usually the latest football match. So this Sunday she sat down and looked at the past and scheduled matches to decide which upcoming matches she wants to watch. To keep it interesting, she only wants to watch matches of teams that still have a chance to win the current tournament. Help her to find out which teams could still win and which cannot.

The tournament consists of a series of matches some of which have already occured. Each match has a winner, there is no draw. In the end, there is either a single team that has the most wins or there will be playoffs between all teams with the (identical) maximal number of wins. Lea considers it possible for a team i to win the tournament if it is possible to assign a winner to each match such that team i has at least as many wins as any other team at the end of the tournament (thus this team reaches the playoffs, if any, or is directly declared winner).

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with two integers, n and m, the number of teams n indexed from 1 to n and the number of matches m that are scheduled to be played. The next line contains n space separated integers w_i, where w_i is the number of wins team i already has. m lines follow describing the matches that are to be played. Line i contains two distinct integers a_i and b_i, the two teams that play each other in match i.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is a space separated list that contains n items, each either "yes" if team i can still win or "no" if not. Each line of the output should end with a line break.

Note: This tournament is not like any real life tournament. In particular there may be teams that play a lot more matches than others.

Constraints

- $1 \leq t \leq 20$
- $2 \leq n \leq 30$
- $0 \leq m \leq 300$
- $0 \leq w_i \leq 1000$ for all $1 \leq i \leq n$

- $1 \le a_i, b_i \le n$ for all $1 \le i \le n$
- $a_i \neq b_i$ for all $1 \le i \le n$

Sample Data

Input

```
2
3 1
1 1 1
1 2

2 2
2 1
1 2
1 2
```

Output

```
Case #1: yes yes no
Case #2: yes yes
```