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

Winter Term 2014
Problem Set 13
22.01.2015

Algorithms for Programming Contests

This problem set is due by

Thursday, 29.01.2015, 6:00 a.m.

Try to solve all the problems and submit them at

`http://icpc.tum.de/judge/`

All students are invited to join this contest. If you do not have an account yet register on the website given above or write a message to conpra@in.tum.de, we have additional accounts. You may work together in teams of two students. Please write a message to us if you want to have a team account.

Sample solutions, statistics and small prices for the winners will be given on Thursday, 29.01.2015, at 12:00 p.m. in room MI 00.08.038.

There will be 6 points awarded for each problem solved for contestants of the practical course. We will only use the usual 28 points for computing the total number of points, everything else is a bonus.

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 you want to have judged

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 us by using the judge's clarification form.

WS14N13A Escaping the Paradox

During a recent archaeological trip, Lea discovered a vast complex of underground tunnels connecting graves of an ancient civilisation. She spent a long time wandering around and drawing a map that captures all the graves and connections. Right now, she is deep below the surface, and has finally reached the last unexplored chamber, which is, disappointingly, completely empty. "Must have been budget cuts" she is thinking, when suddenly a distant feeling of danger makes her feel uneasy. Only an instant later, she knows with absolute certainty the source of the feeling: It is herself. Or, her future self, to be exact.

She does not yet know why she would want to use a time machine to come back here from the future, but basic knowledge of time travel is of course to never interact with yourself from another time. Luckily, she knows exactly where her future self is right now: Only one of the caves is big enough for the time machine. Now she has to figure out how she can escape to the surface without any chance of running into her clone. Her first decision is to always run towards the surface. Of course this problem is not that hard, but Lea also wants to take as many objects of the ancient civilization with her as possible. Luckily, her map includes a list of the objects in each room. How many objects can she take with her on her way to the surface while making sure there is no way to run into herself?

Lea and her future self need the same time to move the same distance and Lea has to take a way so that it is never possible for her and her future self to be in the same location at the same time. In particular Lea has to exit to the surface before her future self can do so. Furthermore, tunnels are undirected for future Lea, but Lea can only move in direction of the surface, that is, to graves with smaller indices.

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 three integers n, m, and g, the number of graves n, the number of tunnels m and the grave g where Lea's future self starts. Graves are indexed from 1 to n where n is the grave where Lea is right now. The following line contains n integers $o_1 \dots o_n$, o_i is the number of objects in grave i. m lines follow describing the tunnels, the j-th line contains three integers x_j y_j l_j. x_j and y_j are either grave indices or 0 (the surface) meaning that there is a tunnel between x_j and y_j of length l_j.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is either the maximal number of objects Lea can take with her on the way to the surface (may be 0) or "impossible" if she cannot escape her future self. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$

- $1 \leq n \leq 500$
- $1 \leq m \leq 5000$
- $1 \leq g \leq n$
- $1 \leq o_i \leq 50$ for all $1 \leq i \leq n$
- $1 \leq l_j \leq 50$ for all $1 \leq j \leq m$
- $0 \leq x_j, y_j \leq n$ for all $1 \leq j \leq m$
- The graph is connected.

Sample Data

Input

```
2
3 3 2
1 1 1
0 1 1
1 3 1
1 2 2

3 3 2
1 1 1
0 1 1
1 2 1
1 3 2
```

Output

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

WS14N13B Beer Pipes

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

- $1 \leq t \leq 20$
- $3 \leq n \leq 1000$
- $1 \leq m \leq 5000$
- $1 \leq a_i, b_i \leq n$ 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
```

WS14N13C Cookies

Most people are quite happy to invite friends. Lea is too, and of course she strives to make the invitees as happy as possible. Sometimes, this proves to be quite difficult. This time she plans to buy cookies for everyone. While this sounds like a simple task, the eating habits of her friends complicate things.

When eating cookies, all friends sit in a big circle. All cookies are poured into a big bowl that is passed around. Each friend has a specific number of cookies that he eats every time he gets the bowl. The bowl starts full at Lea, is passed around each time in the same order, and Lea always eats exactly one cookie whenever she gets the bowl back.

She now wants to buy a number of cookies such that, no matter which of her friends show up, the bowl will end up empty after Lea takes a cookie (Then the bowl is passed around no more). It may be passed around a couple of times, but it should not happen that a friend cannot take his number of cookies or that it returns to Lea empty.

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 an integer n, the number of Lea's friends. The next line contains a space separated list of n integers $c_1, \ldots, c_n$, c_i is the number of cookies her i-th friend eats each time he gets the bowl.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is the minimal number of cookies Lea has to buy to satisfy the constraints above modulo $2147483647 = 2^{31} - 1$. Each line of the output should end with a line break.

Constraints

- $1 \le t \le 20$
- $1 \le n \le 500$
- $1 \le c_i \le 50$

Sample Data

Input

```
2
2
3 5

3
1 1 1
```

Output

```
Case #1: 36
Case #2: 12
```

WS14N13D Ghost

One lazy afternoon, Lea and her friend Bea sit at home. They are both bored out of their minds, so they decide to play a game to pass the time. Since both are too lazy to get up from the sofa and get a boardgame from the cupboard, they settle on the game "Ghost". "Ghost" is a simple word game, where Lea and Bea take turns announcing a single letter to build a word fragment. The rules are as follows: The announced letters have to form the beginning of a valid word. However, the player who completes a word loses.

So for example if Lea starts with "P" and Bea replies with "i", then Lea would lose if she said "g" or "e". She could, however, say "z" (starting to spell "Pizza", for example) and thus continue the game. However, if Lea (or Bea, respectively) did not know any word starting with "Piy", then announcing "y" would be against the rules.

Knowing this day would come, Lea has already prepared a list of all the words she knows. Smugly, she assumes that Bea does not know any words she does not know herself. Now, Lea and Bea play a few rounds culminating in a "final showdown" round. The loser of the last round has to get up and get a boardgame from the cupboard. Bea is also quite skilled at the game, so Lea wants to know if she can win against Bea even if Bea plays perfectly.

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 consists of two integers n and w, with n being the amount of rounds played and w being the amount of words Lea knows. w lines follow, each containing a word s, consisting of lowercase letters from "a" to "z".

Output

For each test case, output one line containing "Case #i:" where i is its number, starting at 1. Then output four lines, corresponding to four different scenarios. In the i-th line, print "victory", if Lea can win the last round n according to the i-th scenario, and "defeat" otherwise.

- Scenario 1: Lea begins in round 1. The winner of round k begins in the next round $k + 1$.
- Scenario 2: Lea begins in round 1. The loser of round k begins in the next round $k + 1$.
- Scenario 3: Bea begins in round 1. The winner of round k begins in the next round $k + 1$.
- Scenario 4: Bea begins in round 1. The loser of round k begins in the next round $k + 1$.

Constraints

- $1 \leq t \leq 50$
- $1 \leq n \leq 1000$
- $1 \leq w \leq 15000$
- $1 \leq |s| \leq 300$

Sample Data

Input

```
1
2 5
chilling
wicked
haunting
spooky
fortress
```

Output

```
Case #1:
victory
defeat
defeat
victory
```

WS14N13E Portals

Futureland is a scientifically quite advanced land. They have figured out how to create long range portals that transport people instantly from one location to another. Only two things are needed to open a portal: A portal crystal (which is destroyed in the process) and two portal portal guns (one at the origin and one at the destination). After the discovery, the government built a portal gun in each city and started selling portal crystals in government-controlled crystal shops (conveniently located right next to the portal gun). After a portal is opened it must be immediately used, after its use it closes and a new portal crystal is needed to open it again.

There are multiple types of portal crystals, each type has limits as to which cities it can connect with a portal. Each city has exactly one crystal shop that sells some types of crystals. By government regulations, each person is only allowed to carry one crystal per type at a time.

Lea is in Futureland right now because she wanted to see the portals for herself. There is even a portal museum that depicts the development process and physical explanations. Lea would like to visit that museum, but the complicated portal crystal system is troubling her. What is the cheapest way to get to the museum?

All Lea has is a list of the crystal sellers including their offered crystal types and prices. She must now find a way to travel to the museum as cheap as possible.

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 and m, the number of cities n (indexed from 1 to n) and crystal types m (indexed from 1 to m).

n lines follow describing the crystal vendors. The i-th line contains m integers $p_{i,1} \dots p_{i,m}$ where $p_{i,j}$ is the price for which the crystal vendor in city i sells crystal j. A price of 0 means the vendor does not sell this crystal.

The next m lines describe the portal crystals. The i-th line starts with an integer k_i followed by $2 \cdot k_i$ integers $a_{i,1} \dots a_{i,2 \cdot k_i}$ meaning that crystal i can connect city $a_{i,1}$ to city $a_{i,2}$, city $a_{i,3}$ to city $a_{i,4}$ and so on. Note that these connections are directed.

The last line contains two integers x and y meaning that Lea is in city x and the museum is in city y.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is the minimum amount Lea has to pay for portal crystals to get to the museum or "impossible" if Lea cannot get to the museum. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 200$
- $1 \leq m \leq 10$
- $0 \leq p_{i,j} \leq 50$ for all $1 \leq i \leq n$, $1 \leq j \leq m$
- $1 \leq k_i \leq n$
- $1 \leq a_{i,j} \leq n$ for all $1 \leq i \leq n$, $1 \leq j \leq k_i$
- $1 \leq x, y \leq n$

Sample Data

Input

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

3 1
3
0
0
2 1 2 2 3
1 3
```

Output

```
Case #1: 6
Case #2: impossible
```

WS14N13F Rescue Mission

Back at home, Lea watches a suspense-packed film about an ancient clan of ninjas. Soon, the main character, "Thunderfist" Shen, is in dire trouble: all other ninjas of his clan have been captured by the evil ninja Karai. Now, he has to infiltrate the enemy base to free them.

The enemy fortress is a twisted maze of winding corridors and dark alleyways, overshadowed by looming watchtowers. Scattered throughout the enemy base are several dungeons where his friends are kept. Whenever a ninja reaches one of the dungeons, he can free all ninjas there. When freed, they either stay where they are and go into hiding or an arbitrary amount of them helps to free the other ninjas. Once all ninjas have been freed, they can rise up and kill all the guards, but until then they have to stay undetected by all means. So, to move through the evil fortress, a ninja has to distract the guards along the way or silently sneak past them. Remember, ninjas always work alone, so if two ninjas sneak through the same corridor, both have to distract the guards independently.

Every time one of the ninjas has to sneak past a guard, there is a chance that the guard raises an alarm and the ninjas' plan is foiled.

Lea wants to know if she can plan the rescue better than Shen did in the film. Can you tell her the minimum total amount of times the ninjas have to sneak past guards undetected to free all captives?

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 consists of a line containing three integers n, m, and d, where n is the number of rooms in the enemy base (indexed from 1 to n), m the number of connections between those rooms and d is the amount of dungeons. Shen starts in room 1. d lines follow. The i-th line contains an integer v_i which denotes that room v_i is a dungeon that contains captured ninjas. m lines follow. The j-th line contains three integers v_j, w_j and g_j. v_j and w_j each denote a room in the enemy base, g_j is the amount of times a ninja has to sneak past a guard to move along that corridor.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is the minimum amount of times the ninjas have to sneak past guards. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 50$
- $1 \leq d \leq n \leq 500$

- $1 \leq m \leq 125000$
- $1 \leq v_i, v_j, w_j \leq 500$ for all $1 \leq i, j \leq d$
- $1 \leq g_i, g_j \leq 100$
- The graph is connected.
- There are more than d captives in each dungeon.

Sample Data

Input

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

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

Output

```
Case #1: 6
Case #2: 8
```

WS14N13G Story Time

On another boring afternoon a few months ago, Lea decided to write a fantasy book which she will call "Game of Kings", an epic tale about kings and queens, bishops and knights. Oh, and magic.

She has spent many nights imagining what would happen to all the awesome characters she would describe. Now, she just needs to compile the list of events into one book. For every character, she wrote a list of chapters which are centered around him or her. These have to happen in a fixed order.

Also, some chapters have a specific ordering. For example, the scene in which Queen Lena, nicknamed the "Black Queen", has a nervous breakdown over a letter detailing the abduction of her brother, a famous knight, has to happen after the description of the abduction from her brothers point of view. Lastly, two sequential chapters should not be centered around the same character.

Can you tell her in how many possible ways she can arrange all the scenes she imagined?

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 consists of two integers n and m, where n is the number of characters (indexed from 1 to n) and m is the number of dependencies between chapters. n lines follow. The i-th line consists of a single integer a_i, the amount of chapters that are centered around character i. m lines follow. The j-th line consists of four integers c_1, p_1, c_2, and p_2 and means that the p_1-th chapter centered around character c_1 has to happen before the p_2-th chapter centered around c_2.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is the amount of possible orderings of the chapters such that:

- Sequential chapters are about different characters.
- Chapters about a single character are in a fixed order.
- All m inter-chapter-dependencies are fulfilled.

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

Constraints

- $1 \leq t \leq 50$

- $1 \le n \le 6$
- $0 \le m \le 15$
- $1 \le c_i \le a_i \le 15$
- $p_1 \ne p_2$
- $1 \le p_i \le n$
- There are at most 13 chapters.

Sample Data

Input

```
2
3 0
2
2
2

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

Output

```
Case #1: 30
Case #2: 570
```

WS14N13H Travel Guides

Lea plans to visit all important sights on earth at some point. Of course, this means she will need travel guides that contain information for each sight. She found one specific guide seller, Information Overload, which sells guides ordered by topics: There are guides which contain all the museums (yes, all of them!), others contain all the mountains, some even combine multiple topics like "The ultimate hitchhiker's guide to museums and pyramids". Lea now has to choose which of the guides she wants to buy to cover all topics. She does not neccesarily need the optimal solution, but if there are n topics and the optimal solution uses k guides, she wants to buy at most $k \cdot (1 + \ln n)$ guides where ln is the natural logarithm. Can you give her such a solution?

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 an integer n, the number of different guides. The following n lines describe the guides. The i-th line starts with a number c_i, the number of topics featured in guide i, followed by c_i strings, the topics in guide 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 of the indices of the guides (indexed from 1 to n) that Lea should buy. Your solution will be considered correct if it is within the bounds given above and covers all topics. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 500$
- $1 \leq c_i \leq 500$
- All strings are combinations of at most 20 lower and upper case letters.
- No two strings in the same line are identical.

Sample Data

Input

```
2
3
1 Museums
1 Mountains
2 Museums Pyramids

5
2 Museums Mountains
2 Pyramids Churches
2 Museums Pyramids
1 Mountains
1 Churches
```

Output

```
Case #1: 2 3
Case #2: 1 2
```

WS14N13I Gingerbread

Lea loves cakes and spent her weekend baking some of them. She also made a gingerbread man as a present for her friend Bea. They do not meet often since Bea lives in another part of the country, but from time to time they send letters or small gifts to each other. This time, the gingerbread should be the gift.

When Lea brought it to the post office the officer complained about the gift's weight. The weight is just above the limit for a reasonable price, so Lea has to remove some of the gingerbread. Since she baked a nice (convex) figure, she wants to remove a little from the corners of the gingerbread in the following way: For each edge she marks where $\frac{1}{m}$ of the edge length is (from both directions) for an integer m. Now, there are two markers next to each vertex and she cuts the vertex along the line between these markers. But is this already enough?

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 and m where n is the number of vertices of the gingerbread man and m is the integer given above. n lines follow describing the vertices. The i-th line contains two integers x_i and y_i describing the coordinates of the i-th vertex.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is the ratio between the removed area and the total area of the gingerbread before cutting it. The number should be printed as a rational, e.g. as a/b for two integers a and b. The rational does not need to be simplified. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $3 \leq n \leq 1000$
- $2 \leq m \leq 1000$
- $0 \leq x_i, y_i \leq 1000$ for all $1 \leq i \leq n$
- The gingerbread will always be convex.
- The points will be given in clockwise or counter-clockwise order.

Sample Data

Input

```
2
4 2
1 0
1 1
0 1
0 0

3 10
20 0
0 20
1 1
```

Output

```
Case #1: 1/2
Case #2: 3/100
```

WS14N13J Cable Car

Lea is a great fan of wintersports. She always follows the winter olympics on TV and just loves to go skiing herself. This year, she booked a room in an expensive hotel in a very exclusive ski resort called “Slippery Slopes and Hills”. One evening at the Après-Ski-Party, she met an interesting man - the architect who planned all the cable cars taking the tourists up the mountain. They talked for a bit and he described his latest problem to her.

The ski resort is trying to build a new cable car up a glacier. Through complicated computation, they even found out exactly how many posts are needed to support the cable car. Since glaciers move (albeit really slowly), the individual posts supporting the cable car have to be spaced as far apart from each other as possible. Additionally, the cable car should also span a canyon in the middle of the route. Now the architect is hard at work, trying to figure how to place the posts. Lea, who is always on the lookout for interesting problems, tells you about it. Can you help the architect?

Input

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

Each test case consists of a single line of four integers d, p, u, and v, where d is the length of the route (going from 0 to d) of the cable car, p is the amount of posts that should be placed, u is the beginning point of the canyon and v the end point. Posts may be placed anywhere between 0 and d, i.e. exactly on 0, d, u, and v, but not in between u and v.

Output

For each test case, output one line containing “Case #i: x” where i is its number, starting at 1, and x is the maximal minimum distance between two posts that can be achieved with an absolute error of up to 10^{-4}. This means the maximum x such that the architect can place all the posts and no two posts are less than x apart. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq d \leq 1000000$
- $2 \leq p \leq 2000000$
- $0 \leq u \leq v \leq d$

Sample Data

Input

```
4
2 3 1 2
3 3 0 1
9 10 5 6
9 10 5 7
```

Output

```
Case #1: 1.0
Case #2: 1.5
Case #3: 1.0
Case #4: 0.8333333333
```