Algorithms for Programming Contests
SS20 - Week 05

Chair for Foundations of Software Reliability and Theoretical Computer Science,
TU München

Mikhail Raskin, Tobias Meggendorfer, Stefan Jaax, A.R.Balasubramanian,
Christoph Welzel

Welcome to our practical course! This problem set is due by

Tuesday, 26.05.2020, 6:00 a.m.

Try to solve all the problems and submit them at

`https://judge.in.tum.de/conpra/`

This week's problems are:

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

Problem	A	B	C	D	E
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.

This page is intentionally left blank.

Problem A
Railroad Network

The preparation to Lea's vacation in Templonia is under way, and after coming back from the Money Exchange, she plans her trips. Sadly enough, the Templonian airlines have very bad reputation. The Templonian people are too much into spirituality and temple architecture to bother about routing baggage from one airport to another. This means that on many occasions a traveler will be sent to its destination, but his or her baggage will not, and will be routed to another airport in Templonia instead. Unfortunately, there is no way around such an airline. Lea's first destination is the capital Tempolis. However, from experience one knows that it is highly probable that the baggage arrives at Starta. Now, the baggage is forwarded to Tempolis by trains, because the domestic railroad network is quite developed. Planning ahead, Lea wants to find a way to forward all the baggage from her plane from Starta to Tempolis. Can you help her find the solution such that the maximal amount of baggage is transported from Starta to Tempolis?

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 cities n (indexed from 1 to n) and the number of railroad connections m. m lines follow describing the railroad connections, each containing three integers a_i, b_i and w_i where a_i and b_i are the endpoints of the connection and w_i describes the maximal baggage weight on this connection. Note that connections are undirected, i.e., if there is a connection from city a_i to city b_i, then there is one in the opposite direction as well.

Assume that baggage can be split up in integral parts as much as you like.

City 1 is considered to be Starta, city n is Tempolis.

Output

For each test case, output one line containing "Case #i:" where i is its number, starting at 1, and the maximal weight of the baggage that can be sent from Starta to Tempolis or "impossible" if there is no path between Starta and Tempolis. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $2 \leq n \leq 500$
- $1 \leq m \leq 5000$
- $1 \leq a_i, b_i \leq n$ for all $1 \leq i \leq m$
- $1 \leq w_i \leq 1000$ for all $1 \leq i \leq m$

Sample Input 1

```
2
4 5
1 2 8
1 3 5
2 3 6
3 4 12
2 4 1

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

Sample Output 1

```
Case #1: 12
Case #2: 11
```

Sample Input 2

```
5
7 3
5 6 1
1 4 8
4 2 7

2 5
1 2 8
1 2 2
1 2 3
2 1 3
1 2 10

5 8
1 5 8
1 3 9
3 1 9
5 3 1
5 4 7
3 5 7
4 2 10
5 4 3

4 4
3 4 4
1 4 9
1 4 2
3 2 9

2 2
1 2 2
1 2 7
```

Sample Output 2

```
Case #1: impossible
Case #2: 26
Case #3: 16
Case #4: 11
Case #5: 9
```

Problem B
Fine Dining

After solving problems all day, every day for the last few weeks, Lea decided to calm down a little and celebrate. She invited all of her friends and together they went to have a nice evening out at a very fancy local restaurant - the "Highlander Restaurant".

A waiter in fine apparel approached them and explained the menu. Naturally, the group began discussing what they wanted to order from the extravagant menu. Soon, everybody settled on a main dish and a beverage.

The waiter also explained that there was a special for groups - the "Highlander Menu". In this special menu, the group would be given exactly one of everything on the menu, i.e. one Pizza Salami, one Lasagna, one bottle of Merlot, one bottle of craft beer, etc..

Lea, intrigued, loved the idea. But since everybody had already chosen their favorite food and beverage, how many people would actually get what they wanted?

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 single line containing three integers $n\ m\ b$, where n is the number of people ordering (Lea included), m is the number of different main dishes on the menu and b is the number of different beverages on the menu (numbered from 1 to m and 1 to b, respectively). n lines follow detailing the preferences of Lea and her friends. The i-th line contains two integers $m_i\ b_i$, specifying that friend i wants to eat main dish m_i and drink beverage b_i.

Output

For each test case, output one line containing "Case #i: y" where i is its number, starting at 1, and y is the maximum number of friends that could get their favorite main dish and beverage if Lea ordered the "Highlander Menu".

Constraints

- $1 \leq t \leq 20$
- $1 \leq n, m, b \leq 500$
- $1 \leq m_i \leq m, 1 \leq b_i \leq b$ for all $1 \leq i \leq n$

Sample Input 1

```
2
2 2 2
1 2
2 1

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

Sample Output 1

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

Sample Input 2

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

10 9 4
7 3
5 2
4 1
7 3
7 3
3 4
1 3
4 1
2 1
3 2

2 3 6
3 1
1 4

1 7 6
1 1

6 9 1
7 1
2 1
3 1
9 1
7 1
5 1

1 9 3
9 2

5 10 8
5 3
10 1
10 2
10 5
9 8
```

Sample Output 2

```
Case #1: 3
Case #2: 4
Case #3: 2
Case #4: 1
Case #5: 1
Case #6: 1
Case #7: 3
```

Problem C
Connectivity

Lea is somewhat addicted to the newest web series "Guardians of the Enigma". Therefore it is important for her to always be connected to the internet to access the next episode as soon as it is released. However, recently, it has often happened that as soon as she wanted to watch it, some problem arised: the battery in her tablet was empty, the Wi-Fi router kept losing the connection, or her desktop did not not boot any more due to the newest system upgrade.

To avoid this problem, she has bought more devices and added even more connections between them. For example, she can use her smartphone, tablet, or laptop for watching, all of which are connected through her Wi-Fi router to the internet. Additionally, she can use her desktop computer connected by a cable to her local server, which then also has direct access to the internet. In this scenario, even if one her devices fails (the smartphone, laptop, router, desktop, or server), she can still access the internet. However, if the router and either the desktop or server breaks, the connection is lost by a failure of just two devices.

Now Lea needs to know how reliable her setup is. For that, she wants to know exactly how many devices have to fail for her to lose her connection to the internet. Fortunately, the connections themselves, e.g. the cables, will never fail, and also the internet endpoint is always working. Given the devices and their connections, as well as the devices Lea can use and the devices connected to the internet, can you tell Lea this number?

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 two integers n, the number of connection points, and m, the number of connections. m lines follow. The j-th line contains two integers a_j b_j, meaning that connection point a_j and b_j are connected. All connections are usable in both directions. The connection points are indexed from 1 to n, where number 1 is Lea, number n is the internet endpoint, and all numbers strictly between 1 and n are intermediate devices. The devices connected to 1 are the devices Lea has access to, and the devices connected to n are the devices with direct access to the internet endpoint.

Output

For each test case, print a line "Case #i: x", with i being the number of the test case, starting at 1, and x being the minimum amount of intermediate devices (excluding Lea and the internet endpoint) that have to fail for Lea to have no connection to the internet any more.

Constraints

- $1 \leq t \leq 20$
- $3 \leq n \leq 400$
- $2 \leq m \leq 100000$
- $1 \leq a_j, b_j \leq n$ for all $1 \leq j \leq m$
- $a_j \neq b_j$ for all $1 \leq j \leq m$
- There will always be at least one possible path from 1 to n.
- There will never be a direct connection from 1 to n.

Sample Input 1

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

4 4
1 2
2 3
2 4
3 4

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

6 8
1 2
1 3
1 4
2 3
3 6
4 5
4 6
5 6

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

Sample Output 1

```
Case #1: 2
Case #2: 1
Case #3: 3
Case #4: 2
Case #5: 2
```

Sample Input 2

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

4 5
1 2
1 3
2 3
2 4
3 4

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

3 2
1 2
2 3

9 18
1 2
1 3
1 4
1 5
1 7
2 3
2 4
2 9
3 4
3 9
4 5
4 6
4 9
5 7
6 8
6 9
7 8
8 9
```

Sample Output 2

```
Case #1: 3
Case #2: 2
Case #3: 2
Case #4: 1
Case #5: 4
```

This page is intentionally left blank.

Problem D
Football Champion

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

Sample Input 1	Sample Output 1
2 3 1 1 1 1 1 2 2 2 2 1 1 2 1 2	Case #1: yes yes no Case #2: yes yes

Sample Input 2

```
9
3 3
1 0 1
3 2
3 2
1 3

4 2
1 0 0 1
2 1
4 2

2 1
1 1
2 1

6 3
0 1 0 1 0 1
3 4
6 3
1 3

3 4
0 1 3
1 3
1 2
1 3
2 3

3 0
0 0 0

3 4
2 0 2
2 1
1 2
3 1
2 1

4 3
1 0 2 2
2 3
4 2
4 1

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

Sample Output 2

```
Case #1: yes yes yes
Case #2: yes yes no yes
Case #3: yes yes
Case #4: no no yes yes no yes
Case #5: yes yes yes
Case #6: yes yes yes
Case #7: yes yes yes
Case #8: yes yes yes yes
Case #9: yes no yes yes
```

Problem E
Goat Riders

Last night, Lea watched an interesting documentary about some of the more remote areas of the world. Deep in the mountains, there is still one clan left living according to the old ways.

They are called the "Goat Riders" - and they are living in a very harsh environment. Upon reaching maturity, every member of the clan receives a goat as a lifelong companion. Together, they will climb the mountains in search of food and shelter.

Surviving during the winter is especially hard, since sometimes the snow fills entire valleys and everyone that stays there overnight freezes to death. So if the snow gets too deep, the goat riders have to climb to higher ground - otherwise both goat and rider freeze to death.

According to traditions, the clan has divided his territory into grid sectors (indexes with a row and column). Every goat rider can only move one single grid sector per day, since they still need to search for food. Each grid sector is small enough that there is only enough shelter for one single goat rider to survive there.

The goat riders have survived in this harsh environment for centuries and have learned to read the weather. They have learned to perfectly predict how deep the snow will be throughout the winter.

Given a description of the territory, the starting location of the riders and the height of snow on each day until the start of spring, what is the maximum number of goat riders that can survive the winter?

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 three integers n, k and d, where n is the size of the goat riders territory (the territory is an $n \times n$ grid), k is the number of goat riders, and d is the number of nights the goat riders have to survive.

n lines follow, each containing n integers $height_i$, representing the height of each grid sector of the territory. The first row in the input is row 0, and the last is row $n - 1$. The first column in each row is column 0, and the last is column $n - 1$.

k lines follow, each containing two integers r, c, indicating that on day 0 there is a goat rider in row r and column c. No two goat riders will be in the same position.

d lines follow, with the j-th line containing a single integer $level_j$, indicating that during the night from day $(j - 1)$ to day j the snow will rise or fall to level $level_j$. (So the first line will contain the level that the snow will be on the morning of day 1.) The goat riders can still move during day 0. A grid section is considered to be *frozen* if its $height \leq level$ of the snow. All goat riders that start their day on a frozen section freeze to death.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1 and x is the maximum possible number of surviving goat riders.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 100$
- $0 \leq k \leq 100$
- $1 \leq d \leq 24$
- $0 \leq height_i \leq 100$
- $0 \leq level_j \leq 100$

- No goat rider can leave the territory.

Sample explanation

In sample 1, case 1, there is only one single goat rider, starting at $(1, 1)$ at height 1. On the first day, the rider moves to $(2, 1)$ with height 4. During the first night, the snow falls to height 0. During the second and third day, the rider stays at $(2, 1)$ at height 4 and thus survives until the morning of day 4.

In case 2, there are 3 riders. The first one can stay where he is for one night, then go a column to the right while the snow is low and then go another one to the right before the snow rises again. The second rider could do the same: stay for the first night, then go up a row, then go up another row. Unfortunately, the first rider is also at the same grid section, so only one of them survives. The third rider cannot reach the safe peaks higher than 6, so he is doomed anyway.

Sample Input 1	Sample Output 1
2 3 1 3 2 1 2 3 1 3 2 4 2 1 1 0 2 1 4 3 3 4 1 9 1 1 6 1 0 1 1 1 2 4 1 1 9 0 0 1 1 3 0 3 0 6	Case #1: 1 Case #2: 1

Sample Input 2

```
4
5 2 3
4 1 9 6 3
6 1 8 0 3
5 7 4 2 1
1 4 9 2 2
4 3 5 6 4
1 0
4 2
4
0
6

6 5 5
7 0 5 0 6 8
9 4 3 0 7 8
6 1 8 4 6 4
5 1 2 6 4 2
2 7 7 8 2 1
3 2 2 7 7 5
5 4
5 1
4 1
5 3
3 1
3
7
0
3
6

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

9 4 5
3 0 7 3 0 9 4 2 8
0 5 1 2 4 5 2 5 7
5 0 3 0 6 7 0 9 0
4 2 6 0 7 6 4 3 7
1 3 2 5 2 1 2 3 8
9 5 9 0 1 8 8 9 0
3 6 2 0 5 4 1 9 6
1 8 3 3 3 4 8 0 8
2 9 6 0 0 7 0 4 5
1 2
0 2
1 7
6 3
1
6
6
1
1
```

Sample Output 2

```
Case #1: 2
Case #2: 1
Case #3: 1
Case #4: 3
```