Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen (LEA)
Prof. Dr. Harald Räcke
Moritz Fuchs, Philipp Hoffmann, Christian Müller
Chris Pinkau, Stefan Toman

Winter Term 2015/16
Problem Set 3
28.10.2015

Algorithms for Programming Contests

This problem set is due by

Wednesday, 04.11.2015, 6:00 a.m.

Try to solve all the problems and submit them at

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

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.

A Networking

Author: Philipp Hoffmann

Sometimes, Lea has to attend loud and boring parties. In the beginning of such events, everybody wants to say hello to all the other people attending. This usually creates a big mess, so this year a new system was introduced: The transitive symmetric stationary greeting system (TSSGS). The system works as follows: Greetings are now transitive, that is, if Lea greets Ralph and Ralph greets Tom, then Lea is considered to have greeted Tom (thus transitive). Greetings are also symmetric, thus if Lea greets Tom (directly or via transitivity), Tom is considered to have greeted Lea. As to further reduce work, people do not move through the room but simply shout out to persons they want to greet (thus stationary). Still everybody wants to greet everybody (possibly indirectly).

This system of course reduced the work, but now everybody was shouting through the room and it soon got very loud.

Lea wants to do one more optimization: The sound level should be as low as possible while satisfying all the constraints above. A greeting between two people is as loud as the distance between them, the sound level is the sum of all greetings that take place. Help Lea with that problem by providing the lowest sound level 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 an integer n, the number of people at the party, n lines follow. The i-th line consists of n integers $m_{i,j}$ where $m_{i,j}$ is the distance of person i to person j. It is always the case that $m_{i,i} = 0$ and $m_{i,j} = m_{j,i}$.

Output

For each test case, output one line containing "Case #i:" where i is its number, starting at 1.

Starting in the next line, output the greetings that take place, one per line. For each greeting, output its start person i and end person j, separated by one space, such that $i < j$. Furthermore, order your output lexicographically, that is, greeting ab should appear before greeting ij if $a < i$, or $a = i$ and $b < j$. Each line of the output should end with a line break.

If there are multiple ways the greetings can take place with minimal sound level, any of them will be accepted.

Constraints

- $1 \leq t \leq 20$.

- $1 \leq n \leq 150$.
- $0 \leq m_{i,j} \leq 10000$ for all $1 \leq i, j \leq n$
- $m_{i,i} = 0$ and $m_{i,j} = m_{j,i}$ for all $1 \leq i, j \leq n$

Sample Data

Input

```
10
2
0 1
1 0

3
0 1 3
1 0 2
3 2 0

4
0 6 7 4
6 0 5 4
7 5 0 7
4 4 7 0

3
0 3 4
3 0 6
4 6 0

4
0 5 4 3
5 0 7 6
4 7 0 5
3 6 5 0

3
0 3 4
3 0 5
4 5 0

4
0 4 3 5
4 0 5 7
3 5 0 4
5 7 4 0

5
0 6 7 7 3
6 0 5 4 7
7 5 0 7 4
7 4 7 0 7
3 7 4 7 0

3
0 3 3
3 0 5
3 5 0

4
0 4 5 6
4 0 4 7
5 4 0 5
6 7 5 0
```

Output

```
Case #1:
1 2
Case #2:
1 2
2 3
Case #3:
1 4
2 3
2 4
Case #4:
1 2
1 3
Case #5:
1 2
1 3
1 4
Case #6:
1 2
1 3
Case #7:
1 2
1 3
3 4
Case #8:
1 5
2 3
2 4
3 5
Case #9:
1 2
1 3
Case #10:
1 2
2 3
3 4
```

B Technical Difficulties

Author: Philipp Hoffmann

Lea is on vacation and this time, she takes a city tour. She has already planned all the things that she wants to visit. Unfortunately, the public transport provider is not very reliable. Every day there are technical difficulties. In fact, exactly one connection is not usable on each day.

Lea is worried that this might make her trip impossible. For her plan to work, it must be possible to go from every station to all other stations. So from now on, every morning she looks up which connection is not working that day. But she still does not know whether this connection was needed or not! Can you provide her with a list of connections that, if not usable, make it impossible to reach every station from every other station?

Input

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

Each test case starts with two integers n and m, the number of stations and connections in the public transportation network. m lines follow, describing the connections. The i-th line consists of two integers u_i and v_i, the two stations connected by the i-th connection. Connections are undirected.

Output

For each test case, print a line containing "Case #i: x" where i is its number, starting at 1, and x is a sorted, space-separated list of the indices of connections that make Lea's trip impossible if they are not usable. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$.
- $1 \leq n \leq m + 1 \leq 10000$.
- $1 \leq u_i, v_i \leq n$ for all $1 \leq i \leq m$.
- The public transportation network will be connected.
- There is at most one connection between each pair of stations.
- No station is connected to itself.

Sample Data

Input

```
13
4 4
1 2
2 3
3 4
2 4

4 3
1 2
2 3
3 4

3 3
1 2
2 3
3 1

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

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

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

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

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

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

```
4 5

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

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

10 18
1 3
1 4
1 6
1 8
1 9
1 10
2 4
2 10
3 4
3 8
4 5
4 6
4 8
5 7
6 7
6 9
7 10
8 10
```

Output

```
Case #1: 1
Case #2: 1 2 3
Case #3:
Case #4: 3 5
Case #5:
Case #6: 3 4
Case #7: 2 5 6
Case #8: 7
Case #9: 1 4 6
Case #10: 1 6
Case #11: 1 6
Case #12:
```

C Jungle Network

Author: Stefan Toman

On the flight home after her last holidays, Lea's plane crashed somewhere in the jungle. Luckily, everybody on board survived, but they have no food except for what they found in the wreckage of the aircraft (which is not that good since Lea chose a very cheap airline).

The radio set which is built into the plane and some batteries are the only thing they have left to call for help. It turns out there are lots of radio sets in the jungle run by the tribes living there, but their supply of batteries is limited and they therefore don't use them often. It would be great if each village was able to pass messages to each other village. The people there love to help each other and forward messages through the radio network until they reach their recipients. Help the people in the jungle to create such a network by telling them which villages need to establish a radio connection in a way that the power consumption is minimal. In return the villages' inhabitants promised to forward your emergency call out of the jungle.

You will be given the coordinates of all villages owning a radio device. To establish a connection between two villages, each of them needs power equal to the square of their distance. Note that the power consumption is not depending on the number of messages sent. To make it even worse, some of the devices are not in best condition, so the inhabitants also tell you the maximum amount of power the devices may use without breaking. Note that each device may establish several connections, each of them using at most the maximum power level. You will join the network as each village will do, too. Your plane has crashed at coordinates $(0, 0)$ and since you have a very modern device it has no limits in power consumption as long as the recipient is located in the jungle. All villages and the plane have enough batteries remaining to power as many connections with their device as they want, but they want to minimize the total power consumption.

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 n, the number of villages. n lines follow. The i-th line consists of three integers x_i, y_i, and c_i where x_i and y_i are the village's coordinates and c_i is the maximum power level their radio device can consume.

Output

For each test case, print a line containing "Case #i: x" where i is its number, starting at 1, and x is the minimum total power all radio sets consume to set up the network, or the string "impossible" if there is no way to connect all villages.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 1000$
- $-100 \leq x_i, y_i \leq 100$ for all $1 \leq i \leq n$
- $0 \leq c_i \leq 40000$ for all $1 \leq i \leq n$
- There won't be two distinct villages at the same position and no village will be at position $(0, 0)$.

Sample Data

Input

```
11
1
1 1 10

2
0 1 1
0 2 10

1
1 1 1

7
-32 -44 25382
-7 -63 19323
28 95 12389
-56 36 17439
-21 -4 2197
76 95 36065
48 -58 23070

3
30 -94 5076
-39 98 9331
-41 11 9577

5
-46 -31 24927
-66 -48 6370
-46 -49 12710
5 15 38589
-92 88 23578

2
-24 -80 8210
-10 88 36990

2
-86 -83 31777
-80 39 23565

6
19 -73 21839
62 -20 4721
-94 -47 5502
-97 71 37975
-89 -55 35212
-68 -1 22005

6
-37 95 6703
29 -71 28837
-77 -17 15139
-14 31 23199
46 99 23871
-43 -91 247

4
75 -57 35800
66 -25 18198
-43 -16 37936
-1 -47 38088
```

Output

```
Case #1: 4
Case #2: 4
Case #3: impossible
Case #4: 45518
Case #5: impossible
Case #6: 37580
Case #7: 29640
Case #8: 44412
Case #9: 44866
Case #10: impossible
Case #11: 20786
```

D Pick-up Sticks

Author: Christian Müller

Once again, Lea invited Bea over for a nice laid-back afternoon of playing games. Bea almost always lost to Lea in the games they played, so Lea was looking forward to a relaxed time.

However, Bea is sick of losing. She spent the last week practicing all the games she ever played against Lea and as soon as she arrives at Lea's house, she announces that she challenges Lea for the title of "Queen of Games". Intrigued, Lea does not back down from the challenge but announces that she will not hold back and will try to win every single game. (The title does include some nice perks - Lea is looking forward to making Bea call her "Queen Lea, Crusher of Rooks, Bringer of Checkmates and Protector of the King" in front of all her other friends.)

Before moving on to the strategy games, they warm up by playing a game called "Pick-up Sticks" ("Mikado" in German). This is a game where a bunch of wooden sticks are dropped on the table where they end up in a jumbled pile. The player whose turn it is then tries to pick up one stick without moving any of the other sticks. For every stick he can take, he gets a set amount of points. If he fails and moves any other stick, his turn is over, the sticks are rejumbled and the next player can try.

Figure 1: Pick-up Sticks, from `http://a.pragprog.com/magazines/2010-06/images/iStock_000001225226Small__10avt3__.jpg`

Lea is (as usual) quite skilled at this game. As long as there is no other stick that lies directly on top of the stick she is trying to pick up, she will never fail. For a given pile of sticks, can you tell her how many points she will get?

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 m. n is the number of sticks that are still in the game. m is the number of intersections of sticks, where one stick lies directly on top of another one. A line of n space-separated integers $p_1 \dots p_n$ follows where p_i is the point value of the i-th stick. m lines follow. The j-th line contains two integers a_j b_j and means that there is a point where stick a_j lies directly on top of b_j.

Output

For each test case, print a line containing "Case #i: p" where p is the maximum number of points Lea can get.

Constraints

- $1 \leq t \leq 20$
- $2 \leq n \leq 10000$
- $1 \leq m \leq 100000$
- $p_i \in \{2, 3, 5, 10, 20\}$ for all $1 \leq i \leq n$
- $1 \leq a_j \neq b_j \leq n$ for all $1 \leq j \leq m$
- If stick a_j lies on top of stick b_j, then b_j can not lie on top of a_j for all $1 \leq j \leq m$.

Sample Data

Input

```
8
5 4
2 3 3 5 20
1 2
2 5
4 5
5 3

5 4
2 3 3 5 20
1 2
2 5
5 3
3 2

5 4
5 3 3 2 20
1 5
3 4
3 4
4 5

3 3
3 3 3
1 3
2 1
3 2

4 4
20 20 10 10
2 3
2 4
2 1
4 1

5 4
3 20 20 10 20
1 4
1 5
1 2
4 5

9 8
5 10 2 20 5 5 2 10 5
1 2
1 9
2 6
3 2
4 3
4 1
5 2
6 9

5 7
20 20 3 20 20
1 4
1 3
2 3
3 4
3 4
3 4
3 5
```

Output

```
Case #1: 33
Case #2: 7
Case #3: 33
Case #4: 0
Case #5: 60
Case #6: 73
Case #7: 64
Case #8: 83
```

E Travel Trouble

Author: Philipp Hoffmann

Recently, Lea realised that she hadn't seen her uncle, who lives in Chaosville, in a long time, so she decided to visit him right away. The Chaotics (thats what the inhabitants of Chaosville are called) are very simple people who all live alone and far apart from each other in their houses. On earlier visits to Chaosville Lea just drove through the wild to reach her uncle's house. However, in the meantime, the Chaotics invented streets!

Excited with their new discovery, every inhabitant, with the exception of Lea's uncle, built a street from his house to some other house in Chaosville. Despite their chaos, they somehow managed to form a network in which it is possible to reach every house in Chaosville from every other house.

Lea was happy for a moment, until she noticed: The Chaotics had also forbidden off-road driving and she had no idea how to get to her uncle. She only found one piece of help: A printed sheet on which for every two houses in Chaosville, the length of the shortest possible route (in the new road network) between those two houses is listed. Can you help her find the way to her uncle?

Input

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

Each test case begins with a line containing an integer n, the number of houses in Chaosville, n lines follow. The i-th line consists of n integers $m_{i,j}$ where $m_{i,j}$ is the distance of house i to house j. Note that $m_{i,j} = k$ does not neccesarily mean that there is an edge of length k between i and j, but it means that there is a path from i to j of total length k.
It is always the case that $m_{i,i} = 0$ and $m_{i,j} = m_{j,i}$.

Lea always starts at house 1, her uncle lives in house n.

Output

For each test case, print a line containing "Case #i:" where i is its number, starting at 1. In the next line, print the sequence of house numbers h_1 h_2 ... h_k separated by spaces such that $h_1 = 1$, $h_k = n$ and for each pair h_i, h_{i+1} a road exists between house h_i and house h_{i+1}. Furthermore, no house should appear twice in that list.

Constraints

- $1 \leq t \leq 20$.
- $1 \leq n \leq 150$.
- $1 \leq m_{i,j} \leq 150000$.

Sample Data

Input

```
13
5
0 12 12 5 9
12 0 14 7 11
12 14 0 7 3
5 7 7 0 4
9 11 3 4 0

6
0 15 4 19 10 9
15 0 11 4 5 16
4 11 0 15 6 5
19 4 15 0 9 20
10 5 6 9 0 11
9 16 5 20 11 0

4
0 4 13 3
4 0 9 7
13 9 0 16
3 7 16 0

5
0 6 22 13 28
6 0 16 7 22
22 16 0 9 6
13 7 9 0 15
28 22 6 15 0

6
0 12 3 8 17 21
12 0 15 4 5 9
3 15 0 11 20 24
8 4 11 0 9 13
17 5 20 9 0 4
21 9 24 13 4 0

4
0 5 10 14
5 0 5 9
10 5 0 14
14 9 14 0

5
0 5 7 8 12
5 0 12 13 7
7 12 0 15 19
8 13 15 0 20
12 7 19 20 0

4
0 16 8 12
16 0 8 4
8 8 0 4
12 4 4 0

5
0 7 7 16 14
7 0 14 9 7
7 14 0 23 21
16 9 23 0 16
14 7 21 16 0

4
0 18 12 3
18 0 6 15
12 6 0 9
```

```
3 15 9 0

4
0 8 9 3
8 0 17 5
9 17 0 12
3 5 12 0

4
0 11 14 3
11 0 3 8
14 3 0 11
3 8 11 0

4
0 9 5 6
9 0 14 15
5 14 0 11
6 15 11 0
```

Output

```
Case #1:
1 4 5
Case #2:
1 3 6
Case #3:
1 4
Case #4:
1 2 4 3 5
Case #5:
1 4 2 5 6
Case #6:
1 2 4
Case #7:
1 2 5
Case #8:
1 3 4
Case #9:
1 2 5
Case #10:
1 4
Case #11:
1 4
Case #12:
1 4
Case #13:
1 4
```