Algorithms for Programming Contests SS20 - Week 4

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, 19.05.2020, 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
Hiking

Lea enjoys nature a lot, therefore she often goes hiking at the weekend. Last Sunday, she got up early, drove to the foot of a mountain and reached the top just at the right time for lunch. During her dessert, “Apfelstrudel”, she suddenly remembered: She had a very important appointment this afternoon which she was about to miss. In a big hurry, she looked at the map to figure out the fastest way to her car. To her amazement, there were hundreds of hiking trails which crossed multiple times forming thousands of possible routes down! Lea was helpless. Luckily she had her satellite phone with her and called... you! After giving you the list of all the trails she wants to know how far away from her car she currently is. Help her out!

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, where n is the number of intersections (those are numbered from 1 to n) and m is the number of hiking trails. m lines follow. The i-th of those lines contains three integers v_i, w_i and c_i, meaning that intersections v_i and w_i are connected by a hiking trail of length c_i. Hiking trails are undirected, but a pair of intersections may be connected by multiple hiking trails. Lea is currently at intersection 1, her car at intersection n.

Output

For each test case, output one line containing “Case #i: d” where i is its number, starting at 1, and d is the shortest distance of intersection 1 to intersection n. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 1000$
- $1 \leq m \leq 50000$
- $1 \leq c_i \leq 1000$ for all $1 \leq i \leq m$
- $1 \leq v_i, w_i \leq n$ for all $1 \leq i \leq m$

Sample Input 1

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

3 3
1 2 1
1 3 3
2 3 1

3 2
1 2 1
2 3 2

3 2
1 3 5
2 3 4
```

Sample Output 1

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

Sample Input 2

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

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

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

3 2
1 2 1
2 3 5

2 1
1 2 3

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

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

Sample Output 2

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

Problem B
Currency Exchange

Summer is coming soon and Lea wants to travel the world. Since she was a kid, she always dreamed of visiting the Great Temples of Templonia, and this summer her dream shall come true. As Templonians pay with their own currency, the Column, Lea has to go to a bank to exchange currencies. The Column is very rarely exchanged, so she decides to go to the National Bank, which has almost all the currencies of the world available. While looking at the current exchange rates, she may just have discovered a loophole in the system: Changing a currency via several exchanges may leave her with more money than just changing to the desired currency in one step.

Soon, Lea realized that the optimal sequence of exchanges is found by multiplying exchange ratios. Luckily, she remembered a grade school course she had taken on calculus, and knows that $\log(a \cdot b) = \log a + \log b$. Moreover, $\log(a \cdot b)$ is minimal if and only if $a \cdot b$ is minimal. This way, Lea can sum the logarithms of the exchange rates and find the optimal sequence. Can you help Lea write a program to find the best way to change her money into Columns?

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, separated by a space. n denotes the number of currencies (w.l.o.g. the currencies are labeled 1 to n), m denotes the number of possible exchanges between currencies. m lines follow. The i-th line consists of two integers a_i, b_i, and a double c_i, separated by spaces, which means that the i-th exchange gives the rate c_i for changing the currency a_i into the currency b_i, i.e., one can change c_i units of currency a_i into one unit of currency b_i. Note that this does not imply that Lea can change money back from currency b_i to a_i.

The doubles c_i are given with a dot as the decimal symbol.

Lea's current money is given in currency 1, and the Column is represented by currency n.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is one of the three following answers: If Lea can make an infinite amount of money in any currency, then x is `Jackpot`. Otherwise x is either the best (smallest) exchange rate achieved via a sequence of exchanges; or `impossible` if there is neither a possible exchange between the two currencies nor a way to make infinite money. Your output must have an absolute or relative error of at most 10^{-6}.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 500$
- $0 \leq m \leq 5000$
- $1 \leq a_i, b_i \leq n$ for all $1 \leq i \leq m$
- $a_i \neq b_i$ for all $1 \leq i \leq m$
- $0 < c_i < 25$ for all $1 \leq i \leq m$
- For any (ordered) pair of currencies there exists at most one exchange rate.

Sample Input 1

```
2
4 5
1 2 0.6
1 4 2.0
2 3 0.4
3 1 4.5
4 3 0.4

4 5
1 2 0.6
1 4 2.0
2 3 0.4
3 1 3.0
4 3 0.4
```

Sample Output 1

```
Case #1: 2.000000
Case #2: Jackpot
```

Sample Input 2

```
3
8 10
8 2 2.253667193183237
7 3 21.06270642254913
4 1 16.54304400691424
6 2 11.417274520023643
3 8 0.4842040541716247
3 5 24.89220955129525
6 5 3.9879932153224855
1 3 3.3862989246293607
4 3 18.6557548269335
5 2 4.044574982952042

6 4
4 1 5.49865346857874
2 3 1.041528946247336
4 6 15.198096869720814
2 6 20.72926928096376

5 8
4 2 15.234936910338007
3 2 2.783840758662068
2 3 4.34452037823547
1 5 5.149609531472857
5 1 21.634342229523323
5 3 21.628749390916028
5 2 22.244249973539482
5 4 17.64965379097069
```

Sample Output 2

```
Case #1: 1.639660
Case #2: impossible
Case #3: 5.149610
```

Problem C
Supermarkets

It was a long time ago that Lea last saw Peter. She got to know him at school, but now she has not seen him for years. One day she met Peter by chance and he invited Lea to visit him at his new home.

A few days later when Lea wants to leave by car, she suddenly remembers that she forgot to buy a gift. Therefore, she decides to buy a bottle of wine at some supermarket on her way to Peter. She wants to be on time, so the extra way and time needed to buy the wine should be as short as possible. Some of the supermarkets are huge malls where she would need a lot of time to get her wine, some are known for long waiting times and others are very small and perfect for getting just one item. Lea knows the lengths of all roads, the locations of all supermarkets and the time she would need to buy the wine in each store. Where should she buy the wine to reach Peter as fast as possible?

Input

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

Each test case starts with a single line containing five integers n, m, s, a and b. n is the number of cities (labelled city 1 to city n), m is the number of roads and s is the number of supermarkets. Lea lives in city a whereas Peter lives in city b.

Next, there are m lines describing the roads. The i-th line contains three integers x_i, y_i and z_i and implies that there is a road between city x_i and city y_i (which may be used in both directions) for which Lea will need z_i minutes. s lines follow describing the supermarkets. The j-th line contains two integers c_j and w_j describing a supermarket in city c_j where Lea will need w_j minutes to buy the wine. Note that there may be multiple roads between cities as well as multiple supermarkets per city.

Output

For each test case, print a line containing "Case #i: x" where i is its number, starting at 1, and x is the time she needs to go to Peter formatted as "hours:minutes", for instance "5:23" (add leading zeros to the number of minutes if needed) or "impossible" if there is no way to Peter's house.

Constraints

- $1 \leq t \leq 20$
- $2 \leq n \leq 10000$
- $0 \leq m \leq 50000$
- $1 \leq a \leq n$
- $1 \leq b \leq n$
- $0 \leq s \leq n$
- $1 \leq x_i, y_i \leq n$ for all $1 \leq i \leq m$
- $1 \leq z_i \leq 100$ for all $1 \leq i \leq m$
- $1 \leq c_j \leq n$ for all $1 \leq j \leq s$
- $1 \leq w_i \leq 1000$ for all $1 \leq j \leq s$

Sample Input 1

```
10
2 1 2 1 2
1 2 30
1 15
2 20

2 1 0 1 2
1 2 30

5 5 1 4 2
3 5 18
2 5 14
3 1 5
1 2 14
4 3 1
3 100

3 1 2 1 2
1 2 1
2 45
3 72

3 1 0 3 1
2 3 14

5 0 0 3 1

6 0 1 1 3
4 106

7 5 3 7 2
7 7 14
1 1 16
6 3 5
6 2 14
7 6 17
4 119
3 48
2 103

2 0 1 2 1
2 110

4 3 3 4 2
4 2 18
2 4 15
4 4 18
3 60
1 91
4 83
```

Sample Output 1

```
Case #1: 0:45
Case #2: impossible
Case #3: 2:00
Case #4: 0:46
Case #5: impossible
Case #6: impossible
Case #7: impossible
Case #8: 1:29
Case #9: impossible
Case #10: 1:38
```

Problem D
Party Planning

Lea has a lot of things to do. Her birthday is coming up and she wants to host a big party. There are cakes to be baked, decoration to be set up, invitations to be sent, drinks to be cooled, and so on. As Lea likes to plan ahead and dislikes stress very much, she wants to find a perfect schedule to finish all her prep work as quickly and as calmly as possible. Luckily for her, she has lots of close friends (as many as she needs) that propose to help her set up the perfect birthday party. After all, they are the guests and want to enjoy the party as much as Lea does.

Unfortunately, not all needed tasks can be done simultaneously: For example, Lea cannot bake a cake unless all ingredients have been bought before, she cannot decorate the room unless it has been cleaned, and she cannot buy the right amount of drinks until she has checked all the answers on her RSVP cards. But from many years of experience she knows some things. She knows the exact amount of time needed to finish any certain task, and she knows the dependencies of all the needed tasks, i.e. which task has to be done before another can be started. Furthermore, she knows that the first task has to be to write a checklist, and the last task to be completed is to cross off all items on it. But as Lea's friends are not nearly as organized as she is, she needs to present them with an exact schedule so that everyone knows what he or she has to do. Can you help Lea to prepare her party as quickly 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 single line containing an integer n, which denotes the number of tasks (w.l.o.g. numbered from 1 to n). n lines follow. The i-th line consists of a sequence of integers p_i, s_i, and $j_{i,1}, \ldots, j_{i,s_i}$, separated by spaces. p_i denotes the number of time units needed to finish task i, s_i is the number of tasks that are direct successors of task i, i.e. tasks that need task i to be finished before they can be started. The sequence $j_{i,1}, \ldots, j_{i,s_i}$ lists all direct successors of task i, in no specific order.

The first task to be done (the source) is task 1, it is a predecessor of all other tasks; the last task (the sink) is task n, it is a successor of all other tasks.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is the total number of time units needed for the prep work.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 1000$
- $1 \leq p_i \leq 1000$ for all $1 \leq i \leq n$
- $0 \leq s_i \leq n-1$ for all $1 \leq i \leq n$
- $i < j_{i,k} \leq n$ for all $1 \leq i \leq n$ and $1 \leq k \leq s_i$

Sample Input 1

```
3
4
5 3 2 3 4
3 1 4
4 1 4
2 0

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

2
3 1 2
1 0
```

Sample Output 1

```
Case #1: 11
Case #2: 20
Case #3: 4
```

Sample Input 2

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

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

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

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

Sample Output 2

```
Case #1: 25
Case #2: 20
Case #3: 25
Case #4: 32
```

Problem E
Change of Scenery

Every day Lea walks to the lab using the same route as it is the shortest way. This is efficient, but over time she has grown slightly bored of seeing the same buildings and trees and junctions every day. So she decides to look for different routes. Of course, she has a lot to do in the lab, and she does not want to sacrifice time. Thus the new way should be as short as the old one. Is there another way that differs from the old one in at least one junction or path?

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 a line containing three integers N M and K, where N is the number of junctions, M is the number of streets and footpaths in Lea's city, and K is the number of junctions she passes every day

The next line contains K integers, the (1-based) indices of the junctions Lea passes every day. The first integer in this line will always be 1, the last integer will always be N. There is a shortest path from 1 to N along the K junctions given.

M lines follow. The i-th of those lines contains three integers a_i b_i c_i and describes a way from junction a_i to junction b_i of length c_i. Connections between junctions are always undirected.

Note that there may be multiple connections between the same two junctions. The shortest path given uses for every pair of successive junctions a and b a path of minimal length between a and b.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is either "`yes`" if there is another way you can take without losing time or"`no`" otherwise.

Constraints

- $1 \leq t \leq 20$
- $1 \leq K \leq N \leq 10\,000$
- $0 \leq M \leq 1\,000\,000$
- $1 \leq a_i, b_i \leq N$ for all $1 \leq i \leq M$.
- $1 \leq c_i \leq 10\,000$ for all $1 \leq i \leq M$.
- If $N \geq 1000$ for some case then this is the only case. ($t = 1$)

Sample Input 1

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

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

Sample Output 1

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