Algorithms for Programming Contests
WS18 - Week 13

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

Tobias Meggendorfer, Philipp Meyer,
Christian Müller, Gregor Schwarz

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

Wednesday, 05.02.2020, 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:

You may work together in teams of two students.

Six points are awarded for each problem, resulting in 60 points in total. However, only 40 of them will count towards the total number of points, and the additional 20 points will be counted as bonus points.

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.

Problem A
Fast Food

Lea wants to meet her friend Bea at a new restaurant near the campus of the Thomas Underwood University (TUM). Since most of the customers are math students, the owner decided to print the place cards as follows: Each card contains the row a and column b of the table (tables are ordered in a perfect grid) as well as the greatest common divisor, largest common multiple and of course the prime factors of a and b.

Since simply telling your friend at which table you sit would be boring, some customers have invented the following fun game: Instead of telling your friend the exact table-coordinates, you send the gcd of the table you are sitting at as well as the gcds of a few tables to the right. I.e. if you are sitting at table (a, b), you will send $gcd(a, b), gcd(a + 1, b), \ldots, gcd(a + k, b)$ for some k. Your friend then has to figure out at which table you might sit. Because there might be multiple solutions, it has become a custom to tell the friend the right coordinates once he has come up with one correct solution.

Because most of the students communication is performed using Twottr, a short messaging service which allows only 39 letters per message, a and b can each have at most 19 digits.

Of course Bea takes part in this game and has sent Lea some gcds. Can you help Lea figure out where Bea might sit?

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 k, the number of gcds. The next line contains k numbers $a_0, \ldots, a_{k-1}$, where $gcd(a, b) = a_0$, $gcd(a + 1, b) = a_1, \ldots, gcd(a + k - 1, b) = a_{k-1}$.

Output

For each test case, output one line containing "Case #i: a b" where i is its number, starting at 1, and a, b are integers s.t. $gcd(a, b) = a_0$, $gcd(a + 1, b) = a_1, \ldots, gcd(a + k - 1, b) = a_{k-1}$ and $1 \leq a, b \leq 10^{19}$.

Constraints

- $1 \leq t \leq 20$
- $3 \leq k \leq 1000$
- $1 \leq a_i \leq 10^{19} - k$

Sample Input 1	Sample Output 1
2 4 3 2 1 6 4 1 2 3 4	Case #1: 3 6 Case #2: 1 12

Sample Input 2

```
17
3
5 4 3

5
4 5 6 7 8

10
10 3 2 1 6 5 2 3 2 11

7
3 2 1 30 17 2 3

9
7 2 3 2 5 6 1 14 9

10
3 14 11 12 1 2 3 4 7 6

6
84 1 2 3 4 11

6
2 3 14 5 6 1

4
18 7 10 3

10
10 3 2 23 6 5 2 3 2 1

10
12 5 2 3 8 7 30 1 4 3

10
6 5 2 3 2 19 30 1 2 3

10
5 2 9 2 1 30 11 2 3 2

10
10 3 28 1 6 5 4 3 2 7

10
3 2 5 6 7 2 3 10 1 6

10
30 19 2 3 2 5 6 1 2 3

10
2 3 2 13 30 1 2 3 2 5
```

Sample Output 2

```
Case #1: 55 60
Case #2: 4 840
Case #3: 200 330
Case #4: 387 510
Case #5: 91 630
Case #6: 405 924
Case #7: 336 924
Case #8: 152 210
Case #9: 468 630
Case #10: 20 690
Case #11: 324 840
Case #12: 204 570
Case #13: 115 990
Case #14: 110 420
Case #15: 3 210
Case #16: 360 570
Case #17: 296 390
```

Problem B
Sending Money

Lea and her friends often go out together and usually one person pays the bill for the rest. Since it is extremely tedious to balance the resulting debts instantly, Lea quickly wrote a program she called DebtGraph to record all payments. Now, the end of the month is approaching and it is time to settle all debts. Of course, it is of fundamental importance that as little money as necessary is sent around. Unfortunately, Lea is busy with competing in the "Fundamentally Awesome Universe"-contest and thus can't compute the exact transactions necessary to settle all debts this way. Can you help her find an optimal set of transactions settling all debts fairly, i.e. after these transactions everybody has exactly the amount of money which he would have if the debts were settled instantly?

Hint: You do not need to minimize the number of transactions, only the total sum of money sent with these transactions.

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 firends and recorded debts between them, respectively. m lines follow, describing the debts. The i-th line contains three integers x_i, y_i, and c_i, denoting that a debt of x_i owing c_i money to y_i is recorded.

Output

For each test case, output a line containing "Case #i: a" where i is its number, starting at 1, and a is the minimal amount of money that needs to be sent in order to settle all debts. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n, m \leq 10^5$
- $1 \leq a \leq 10^8$

Sample Input 1

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

Sample Output 1

```
Case #1: 5
1 3 5
```

Sample Input 2

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

Sample Output 2

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

This page is intentionally left blank.

Problem C
Round Fence

Lea is fond of wildlife. She wants to protect the nests of the rare endemic wild birds whose natural habitat is under threat from the hunger of the common domestic least-conservation-concerns cats. Her plan includes building a cat-proof round fence enclosing all the trees with the birds' nests. Of course, protecting the wildlife is urgent, so the fence should be built as soon as possible — and the shorter it is, the faster it is to build!

Compute the minimum possible radius of a round fence sufficient for Lea's environmental goals.

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 one integer N. N lines follow, each containing three integers x_i, y_i, and r_i, representing the coordinates (x_i, r_i) and the radius r_i of the i-th tree.

Output

For each test case output a line containing "Case #i: r", where r is the minimum radius of the fence enclosing all the trees. The radius should have the absolute error at most 10^{-3}.

Constraints

- $1 \leq t \leq 20$
- $1 \leq N \leq 1000$
- $1 \leq |x_i|, |y_i|, r_i \leq 10^6$

Trees can overlap.

Sample Input 1	Sample Output 1
1 5 10 11 4 10 -12 3 23 0 2 -4 0 1 10 0 2	Case #1: 15

This page is intentionally left blank.

Problem D
Snowball Fight

Winter is a special time. The first real snowstorm hits, the lakes freeze over, the houses become snowcapped, trees have finally shed all their colors and become frosty skeletons. Everything outside becomes quiet.

Well, not entirely…Ever since they were small children, Lea and her friends have always loved this time. Why? Because once a year, the whole town comes together and celebrates the "Snowball Arena: Free-for-all". For a day, the whole town stands still and anyone who is spotted on the streets can be subject to snowball bombardment. It is an event of joy and of tears, where grown men cry, bombarded by endless blizzards of snowballs, thrown by the hands of children. To make it even more fun, the townspeople have dug trenches on the central square so there are now several fronts that can be besieged.

Of course, Lea takes part in all that. Right now, she is lying alone in one of the trenches and is bombarded by one particularly persistent fellow. To avoid being hit, she needs to keep her head low - thus, she cannot spot her attacker directly. All his snowballs came from the same direction however, so to retaliate, she only needs to know how far her attacker is away.

Luckily, she notices something - to the side of the central square, there is a huge new building[1]. There are no visible windows on the front wall, but rather the whole wall of that building is a huge mirror. Noticing that this could give her a substantial combat advantage, she watches out for movements in the mirror. As soon as she spots someone, she wants to lob a snowball over the siege lines and hit that person right in the face. Can you help her with the target computations?

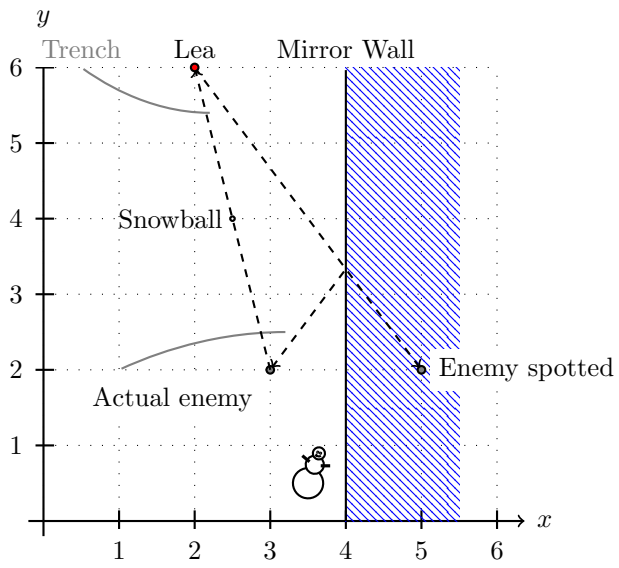

Figure D.1: Illustration of the sample input, case 1.

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 4 lines. The first line contains the integers x_{Lea} y_{Lea}, Lea's coordinates. The second line contains the doubles x_{Snow} y_{Snow}, some in-flight coordinates of one of the snowballs that came from the direction of the attacker. The third line contains the integers x_{Wall1} y_{Wall1} x_{Wall2} y_{Wall2}, the coordinates of the wall. The fourth line contains the doubles x_{enemy} y_{enemy}, the projected coordinates of the enemy she saw in the mirror.

[1] It is owned by some huge tech corporation. Lea forgot the name, but it was something like "Mirror's Ledge".

Output

For each test case, print a line containing "Case #i: x y" where i is its number, starting at 1, and (x, y) are the coordinates of the enemy. Your solution is considered correct if the area is accurate to four decimal places. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $y_{Wall1} = 0$
- $y_{Wall2} = 30$
- All given coordinates are between 0 and 30.
- All points are at least 10^{-4} apart.
- All points are at least 10^{-4} away from the wall.

Sample Input 1

```
1
2 6
2.5 4.0
4 0 4 30
5 2
```

Sample Output 1

```
Case #1: 3.0 2.0
```

Sample Input 2

```
5
14 10
16.5 9.5
20 0 20 30
21.0 9.0

16 15
11.5 10.0
17 0 17 30
27.0 5.0

5 13
10.0 9.5
19 0 19 30
23.0 6.0

19 19
17.0 10.5
20 0 20 30
25.0 2.0

6 17
11.0 13.0
18 0 18 30
20.0 9.0
```

Sample Output 2

```
Case #1: 19.0 9.0
Case #2: 7.0 5.0
Case #3: 15.0 6.0
Case #4: 15.0 2.0
Case #5: 16.0 9.0
```

Problem E
Exam Preparation Revisited

Do you remember the problem "Exam Preparation" in which Lea tried to put as much information as possible on one sheet of paper that she is allowed to use during an exam? Well, Lea needs your help again since she has changed her mind on three things:

- Lea thought there was an unlimited supply of information for each topic, but that turns out to be wrong: For instance, there is no use writing down more than five pieces of information about the birth date of the university's patron Thomas Underwood.
- Since Lea surfs the internet a lot, she did not study as much as she planned to. Now, she needs the perfect cheat sheet with the maximum amount of information.
- The allowed size of the cheat sheets is not as big as expected.

Can you help her again?

Input

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

Each lecture starts with a line containing two integers: m, the number of characters that fit on the allowed cheat sheet, and n, the number of topics covered. n lines describing the topics follow. The i-th line contains three integers p_i, l_i and s_i where p_i is the number of pieces of information available, l_i is the length of a piece of information for this topic and s_i is its score.

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 topics to be added (topic i may appear at most p_i times in this list). The sum of their lengths should be at most m and the sum of their scores should be as big as possible.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 100$
- $1 \leq m \leq 3000$
- $1 \leq p_i \leq 100$ for all $1 \leq i \leq n$
- $1 \leq l_i \leq 100$ for all $1 \leq i \leq n$
- $1 \leq s_i \leq 10000$ for all $1 \leq i \leq n$

Sample Input 1

```
7
10 2
1 3 5
6 1 1

10 3
2 3 7
1 2 8
3 7 5

10 3
1 5 10
3 1 1
3 2 3

6 2
4 6 10
3 3 5

1 6
4 6 10
3 3 5
2 3 2
7 7 23
9 8 17
4 10 8

7 2
2 2 7
5 3 3

8 4
1 4 7
1 2 3
1 2 1
1 2 2
```

Sample Output 1

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

Sample Input 2

```
4
15 3
1 3 50
2 1 96
2 2 9

19 3
2 3 91
2 3 49
1 3 80

20 3
2 1 20
2 1 39
1 1 71

26 3
1 3 21
1 3 98
1 1 10
```

Sample Output 2

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

Problem F
Jeopardy! Fans

Today is the final game of the huge seven-years-long Jeopardy! tournament. Unfortunately, Lea was too busy to participate in the weekly games, so she is in the audience. Each spectator is a fan of one of the three players. Lea didn't have time to ask everyone who supports whom, but she observed some conversations (and participated in some more). Each time she either noticed that two fans are rooting for the same competitor, or that three fans support three different players.

Now it is time to find out how many people have good taste in choosing their heros!

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 begins with a line consisting of two integers n, the number of fans, and m, the number of observations. The fans are numbered from 1 to n, with Lea having the number 1. m lines follow, signaling an observation of support for the same or for different players. The i-th line is either `S` x y, when the spectators from the seats x and y support the **same** player, or `D` x y z, where the fans from the seats x, y, and z represent the full **diversity** of all three possible opinions.

Output

For each test case, output one line containing "Case #i: a b" where i is its number, starting at 1, and a and b are the minimum and the maximum number of fans supporting the same player as Lea.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 2000$
- $0 \leq m \leq 10000$
- $1 \leq x, y, z \leq n, x \neq y \neq z$
- The given relations will not be inconsistent.
- The number of partitions into three support groups consistent with the input will not be larger than 1025.

Sample Input 1

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

Sample Output 1

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

Sample Input 2

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

Sample Output 2

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

This page is intentionally left blank.

Problem G
Nightmare

Lea had a horrible nightmare last night: Together with her friend Bea, she was in the castle of Prof. G. Ruesome. The evil professor plans to seize world domination and is about to finish preparations, but Lea and Bea found his secret. Now they need to tell the world about it in order to stop Ruesome. Horrifyingly, the alarm system observed the intruders and the professor sent his guards to catch them.

Lea and Bea want to flee from the castle to the police station, but they do not know where their pursuers are. Therefore, they decide to split up and run for their lives. It is important that at least one of them reaches the police station, so they do not want to use the same roads. It is ok if their ways intersect, but they should not use any common road. Obviously, they want the sum of their paths to be as short as possible. Help them and save the world!

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 four integers n, m, a and b where n is the number of intersections of roads, labeled from 1 to n, m is the number of roads, a is the number of the node where the castle is situated, and b is the number of the node where the police station is situated. m lines follow describing the roads. The i-th line contains three integers a_i, b_i and c_i, meaning that there is a a road from intersection a_i to b_i of length c_i. Each road may be used from both directions, but not by both of them.

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 sum of the lengths of Lea's and Bea's paths or "impossible" if there are no such paths. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 2000$
- $0 \leq m \leq 20000$
- $1 \leq a, b \leq n$
- $a \neq b$
- $1 \leq a_i, b_i \leq n$ for all $1 \leq i \leq m$
- $1 \leq c_i \leq 100$ for all $1 \leq i \leq m$

Sample Input 1	Sample Output 1
3 4 4 1 4 1 2 1 2 4 2 1 3 3 3 2 4 6 8 1 6 1 2 1 1 3 1 2 4 1 2 5 10 3 4 1 3 5 9 4 6 1 5 6 1 2 2 1 2 1 2 3 1 2 4	Case #1: impossible Case #2: 14 Case #3: 7

Problem H
Optimization

One of the largest companies in Gerland is Ziebenz. They have huge projects and countless employees. Operating at such a large scale can be challenging. For example, one of their current projects contains m independent jobs and involves n workers. The manager of the project has accurately calculated how much working time each part of the project requires. It turns out that the *execution time* for each job is fixed and doesn't depend on the employee who will perform it (all workers are equally talented MUT graduates). After computing the execution times, the manager has randomly (don't judge him, he's a former MUT-LWB student!) assigned all m jobs to n workers.

Of course, it has turned out that some workers will need much longer to complete their jobs. Now the manager wants to improve the situation: he's going to pick up two different workers, and for each worker choose one of jobs assigned to them. Afterwards, he's going to swap the two jobs between the both workers (i.e. assign the first workers' job to the second worker, and vice versa). We'll cal such operation *optimizing*, if the maximum of the overall working times of both workers has decreased after performing the operation.

Let's look at an example. Imagine that the project consists of 5 jobs with execution times of $3, 6, 4, 8$ and 2, and there're 3 workers involved in this project. The initial random assignment is: the first worker gets jobs 1 and 2 (overall working time: $3 + 6 = 9$), the second worker — job 4 (overall working time 8), and the third — jobs 3 and 5 (overall working time $4 + 2 = 6$). Now, if we assign the first job (initially assigned to the first worker) to the third worker, and the fifth job (initally assigned to the third worker) to the first worker, the overall working time of the first worker becomes $6 + 2 = 8$, and of the third worker — $3 + 4 = 7$. Thus, it is an optimizing operation.

You are given the number of employees and the number of jobs, execution times of all jobs, and the initial assignment of the jobs to the workers. Find the number of possible optimizing operations for the given assignment of jobs.

Input

The first line of the input contains an integer $t \leq 20$. t test cases follow, each of them separated by a blank line. Each testcase has the following structure.

The first line contains two integers n and m ($1 \leq n, m \leq 10^5$) — the number of employees and the number of jobs within the project.

The second line contains m integers where the $i-$th number is the execution time of the $i-$th job (jobs are numbered from 1 to m). All execution times are between 1 and 10^9.

The following n lines describe the assignment of the jobs to the workers. The $i-$th of these lines begins with the number k_i — the total number of jobs assigned to the worker i. k_i integers follow — the numbers of the jobs assigned to the worker.

Output

For each test case output a line containing "Case #i: s", where s is the number of possible optimizing operations.

Sample Input 1	Sample Output 1
1 3 5 3 6 4 8 2 2 1 2 1 4 2 3 5	Case #1: 2

Sample Input 2	Sample Output 2
1 2 4 1 2 3 4 2 1 2 2 3 4	Case #1: 4

Problem I
Cannibal Island

Lea, bold and daring as she is, ignores travel warnings. Hence she is held captive on a remote island by a group of cannibals.

Even considering that the cannibals are genuine cannibals, the human-eating savages she is confronted with are of a very malicious kind, for they have devised the following ritual:

The cannibals' leader whispers a number into Lea's ear. Then he leads her to a circle of frightened pigs. The leader points to some position in the circle where a pig is squeaking in agony. "Remember the number k I told you, delicious Lea?", he utters in a lustful voice, "Starting from this position, I pick every kth pig in clockwise manner to be slaughtered until only one pig remains." He smiles eerily, then pauses for dramatic effect, as if to indulge in his cannibalistic desires, before he continues: "You must take the position of one of those pigs. If you are the single individual remaining, we let you free. Otherwise we are going to devour you just like the other pigs."

Can you help Lea find the right position in the circle, so that the mean cannibals let her free?

Input

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

Each test case consists of a single line n k of two space-separated integers n and k, where n is the number of pigs in the circle and k is the number whispered into Lea's ear.

Output

For each test case, print a line containing "Case #i: p" where i is its number, starting at 1 and p is the position of the pig that Lea should replace. The number 1 denotes the starting position the leader points to and the position numbers are assumed to be increased in clockwise order. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n, k \leq 10000$

Sample Input 1

```
2
5 2
8 3
```

Sample Output 1

```
Case #1: 3
Case #2: 7
```

Sample Input 2

```
5
15 3
20 5
5 2
12 4
17 6
```

Sample Output 2

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

This page is intentionally left blank.

Problem J
The Mulk

This afternoon Lea went to an art showing by Cruce Canner in downtown New Tempolis. She did not really like the art and told the artist that she thinks it is pretentious and compared to M.N. Cut's work almost trivial (M.N. Cut is a very famous artist in Templonia). Canner did not take the critique very well, got angry and stormed out the door. A few minutes later, the news showed *The Mulk*, a big yellow raging monster, in front of the art gallery, at which point Lea remembered why the name Cruce Canner somehow sounded familiar.

Realizing that she was the only person in town who knew where *The Mulk* was headed (M.N. Cut's art gallery obviously), she called the town major and offered her assistance in catching *The Mulk*. He told her that they have a lot of roadblocks which can be deployed within seconds at any point in town (this is not their first Mulk-related incident), however, two rules need to be respected:

- Since the roadblocks are very expensive, as few of them as possible should be used - even at the expense of a few cars or houses. In particular this means, that we can let *The Mulk* run around for a bit before trapping him.
- Article 126.7a of the Mulk Defense Act of 2004 states that "in order not to disturb local wildlife too much, all roadblocks must be deployed at the same time".

Can you help Lea figure out how many roadblocks have to be used in the worst case to keep *The Mulk* from reaching M.N. Cut's gallery?

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 begins with two integers n and m. n is the number of road-crossings numbered from 1 to n on the map, m is the total number of roads between road crossings. m lines follow describing the map. Each line contains two integers u and v and describes a road between the crossings u and v.

The Mulk starts at crossing 1 and wants to reach crossing n.

Output

For each test case output one line containing "Case #i: x" where i is its number, starting at 1, and x is the number of roadblocks that need to be used to trap *The Mulk* in the worst case.

Constraints

- $1 \leq t \leq 20$
- $2 \leq n \leq 120$
- $n \leq m \leq 10000$

Sample explanation

In the first sample input we are given 8 crossings with 9 roads in between them. The resulting graph looks this:

Now, we could decide to place roadblocks right away, for example as shown in figure J.2. This uses up 3 roadblocks.

However, if we allow *The Mulk* to run around a bit, we can see, that a better solution is possible: In the first step, he has to decide between going to node 2,3 or 4. When we place the roadblocks after *The Mulk* has made his first step, we can place them for example as shown in figure J.3, which results in only 2 roadblocks being used. Hence the correct answer is 2.

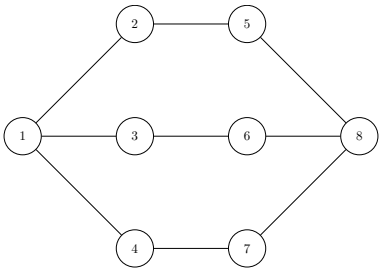

Figure J.1: Graph for the first sample input

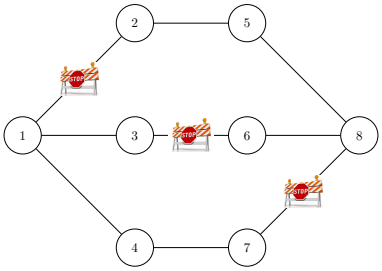

Figure J.2: Non-optimal way of placing the roadblocks.

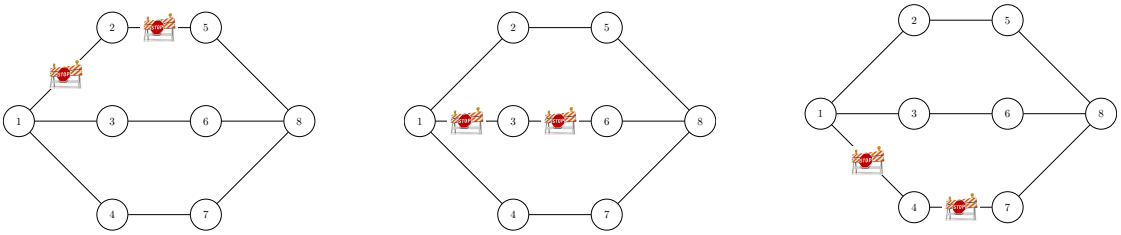

Figure J.3: Optimal roadblocks for all paths from node 1 to node 8.

Sample Input 1	Sample Output 1
3 8 9 1 2 1 3 1 4 2 5 3 6 4 7 5 8 6 8 7 8 5 7 1 5 1 2 1 4 5 3 5 4 5 2 3 4 2 2 1 2 1 2	Case #1: 2 Case #2: 3 Case #3: 2