CS29003 ALGORITHMS LABORATORY
ASSIGNMENT 9
Date: 28th Oct, 2021

Important Instructions

1. **Input:** To be taken from `stdin`
2. **Format of files to be submitted to Moodle and HackerRank:** <ROLLNO>_A9_p1.c(pp), <ROLLNO>_A9_p2.c(pp)
3. You are to **stick to the input output formats strictly** as per the instructions in the respective problems.
4. Write your name, roll number and HackerRank ID at the beginning of each of your programs.
5. Do not use any global variable or STL unless you are explicitly instructed so.
6. Indent and comment your code properly.
7. Please follow all the guidelines. Failing to do so will result in penalty.
8. **There will be partial markings.** Look at the last page for marking scheme.

Greedy on Graph

> "Time. It's more than a linear path. It can be imagined as various *time-nodes* connected via *time-branches*".
>
> The Watcher

Problem 1: Multiverse of Madnessss!

You are Mobius, friend of Loki, the god of mischief. After Sylvie, a Loki's variant, killed Kang The Conqueror, it unleashed multiverse! Now a demon from some other universe is coming to destroy yours. During all this, Loki got back to Time Variance Authority (aka TVA). Loki does not have enough power to defeat the demon. So, he starts looking for some power source in the TVA Library. Miss Minutes helps him and finds an ancient Asgardian scroll. The scroll reads -

For someone who wishes to gain more power shall do the following and have their wish granted:

- ▷ There exists a Sacred timeline S. It has N time-nodes (numbered from 1 to N) **completely** connected to each other via M **bi-directional** time-branches. Hence, $M = N \times (N - 1) / 2$.
- ▷ Start from any time-node t but **must** you visit all the N time-nodes using some time-branches.
- ▷ You are free to visit a time-node more than once. Same applies for time-branches.
- ▷ Time-branches have some length. Gain l amount of power by using a time-branch of length l.

- ▷ The total power, P_t gained will be the summation of powers from all the **distinct** time-branches used. That is, power is gained **only** from the first use of a time-branch. For successive uses of the same time-branch, power gained will be 0.
- ▷ P_t is granted only if all the above rules are followed.

Miss Minutes has arranged a device to time travel called TemPad. She has also feeded in the N time-nodes of the sacred timeline S into the TemPad. Now she asks Loki the **distinct** time-branches he wants to use. Unfortunately, TemPad can only store $N-1$ **distinct** time-branches. Loki being greedy wants to gain as much power as possible to defeat the demon. So, he asks you to calculate the maximum power he can gain by using only $N-1$ distinct time-branches and following the rules in the scroll.

Phew! Now that you have helped your Loki to stop the demon, other Lokis from other universes are also facing the same problem and needs your help. So, Q such Lokis have now come to you. For the i-th Loki L_i, his sacred timeline S_i is almost identical to your sacred timeline S except for time-branch b_i. This time-branch b_i is **longer** in S_i and has a length of x_i. **Since, you have spent too much time in helping your Loki so you have to help these Q Lokis as quickly as possible! That is, you have to come up with an efficient approach for this.**
Note: You are **not** allowed to sort the time-branches so design your algorithm accordingly.

Input Format

- ▷ The first line will have 2 space-separated integers N $(2 \leqslant N \leqslant 300)$ and Q $(1 \leqslant Q \leqslant 20000)$.
- ▷ Then M lines follow each having information about a time-branch. Each line has 3 space-separated integers t_u, t_v and l_{uv}. Here, t_u and t_v are time-nodes $(1 \leqslant t_u, t_v \leqslant N)$ connected by a time-branch of length l_{uv} $(1 \leqslant l_{uv} \leqslant 1000)$.
- ▷ Then Q lines follow where, the i-th line describes the changes for Loki L_i. The i-th line has 3 space-separated integers $t_{u,i}$, $t_{v,i}$ and x_i. Here, $t_{u,i}$ and $t_{v,i}$ are time-nodes $(1 \leqslant t_{u,i}, t_{v,i} \leqslant N)$ connected by the time-branch b_i whose new length is x_i $(2 \leqslant x_i \leqslant 10^4)$.

Output Format

- ▷ First line should have an integer denoting the maximum power your Loki can gain.
- ▷ Then Q lines should follow where, the i-th line should have an integer denoting the maximum power Loki L_i can gain.

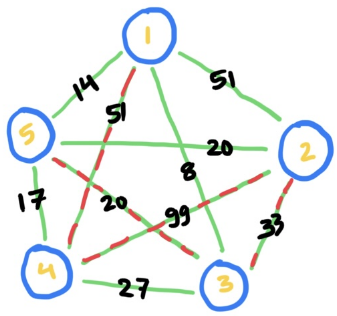

Figure 1: Sacred timeline from sample case. The red dashed time-branches gives maximum power gain for your Loki.

Sample Cases

FILE: *input-problem1.txt*

```
5 4
1 2 51
1 3 8
1 4 51
1 5 14
2 3 33
2 4 99
2 5 20
3 4 27
3 5 20
4 5 17
1 4 53
1 5 26
5 1 19
1 2 67
```

FILE: *output-problem1.txt*

```
203
205
209
203
219
```

Explanation

In the case of your Loki, following time-branches set result in maximum power gain: $\{(1,\ 4),\ (2,\ 4),\ (2,\ 3),\ (3,\ 5)\}$. Refer fig. 1 for visual aid. Other possible time-branches set can be: $\{(1,\ 2),\ (2,\ 4),\ (2,\ 3),\ (3,\ 5)\}$. In both the cases, power gained is 203. There can be other optimal sets but the maximum power gained will be same.

In case of Loki L_1, time-branch between time-nodes 1 and 4 now has length of 53. So, following time-branches set result in maximum power gain: $\{(1,\ 4),\ (2,\ 4),\ (2,\ 3),\ (3,\ 5)\}$ which is 205.

In case of Loki L_2, time-branch between time-nodes 1 and 5 now has length of 26. So, following time-branches set result in maximum power gain: $\{(1,\ 4),\ (2,\ 4),\ (2,\ 3),\ (1,\ 5)\}$ which is 209.

Similarly, other Lokis are handled.

FILE: *input0-problem1.txt*

```
3 1
1 2 51
2 3 33
1 3 8
1 3 39
```

FILE: *output0-problem1.txt*

```
84
90
```

Explanation

In the case of your Loki, following time-branches set result in maximum power gain: $\{(1,\ 2),\ (2,\ 3)\}$. Power gained is 84.

In case of Loki L_1, time-branch between time-nodes 1 and 3 now has length of 39. So, following time-branches set result in maximum power gain: $\{(1,\ 2),\ (1,\ 3)\}$ which is 90.

Bonus Problem: Reset the Timelines

You are a Minuteman at the TVA. Judge Ravonna Renslayer has asked you to reset some branched timelines. A timeline has N time-nodes and M time-branches. There may be multiple time-branches between two same time-nodes. There are rules for resetting a timeline:

- ▷ You have to use some or all of the time-branches from the given M time-branches.
- ▷ Reset charges when placed at a time-node say t, will completely reset all the reachable time-nodes, including t.
- ▷ Cost of resetting a timeline is equal to the summation of lengths of each used time-branch.

Each reset charge has a fixed cost C and can be placed at any time-node. You have to reset all the N time-nodes by minimising the overall cost. Where, overall cost is the total cost of all the reset charges used + total cost of resetting the timeline. If two scenarios have the same minimal overall cost, then you should choose the one with **less** number of reset charges used; as the supply is short. Refer to sample case for clarity. You have to report the overall cost incurred and the number of reset charges used to the Finance department.

There are T such timelines that needs resetting.

Note: It is not necessary to use all the time-branches. Only use the time-branches that helps minimise the overall cost.

Input Format

- ▷ The first line contains the number of timelines T ($T \leqslant 15$).
- ▷ Each timeline case starts with 3 space-separated integers N ($1 \leqslant N \leqslant 10^4$), M ($0 \leqslant M \leqslant 10^5$) and C ($1 \leqslant C \leqslant 10^4$).
- ▷ Then M lines follow having information about time-branches. Each line has 3 space-separated integers t_u, t_v and l_{uv}. Here, t_u and t_v are time-nodes ($1 \leqslant t_u, t_v \leqslant N$) connected by a time-branch of length l_{uv} ($1 \leqslant l_{uv} \leqslant 10^4$).

Output Format

- ▷ Output exactly T lines, one for each timeline. Each line should have 2 space-separated integers O and R, where O is the minimum overall cost of resetting that timeline so that all the time-nodes are reset and R is the number of reset charges used. As mentioned earlier, if there are multiple scenarios with minimal overall cost, choose the one that uses **less** reset charges.

Sample Case

FILE: *input-problem2.txt*

```
2
5 8 20
1 3 8
1 4 51
1 5 14
2 4 19
3 4 20
5 1 22
3 5 27
4 5 47
4 4 100
1 2 10
4 3 12
4 1 41
2 3 23
```

FILE: *output-problem2.txt*

```
81 1
145 1
```

Explanation

For Timeline 1, cost of placing a reset charge is C = 20. Using the time-branches $\{(1,\ 5),\ (2,\ 4),\ (1,\ 3),\ (4,\ 3)\}$ and 1 reset charge at any time-node, we can reset the whole time-line. In this case, overall cost $= (1 \times C) + (14 + 8 + 19 + 20) = 81$. Refer fig. 2 for visual aid. Using time-branches $\{(1,\ 5),\ (2,\ 4),\ (1,\ 3)\}$ and 2 reset charges at time-nodes 2 and 3 also results in same overall cost of 81. So, we choose the one with less number of reset charges used.

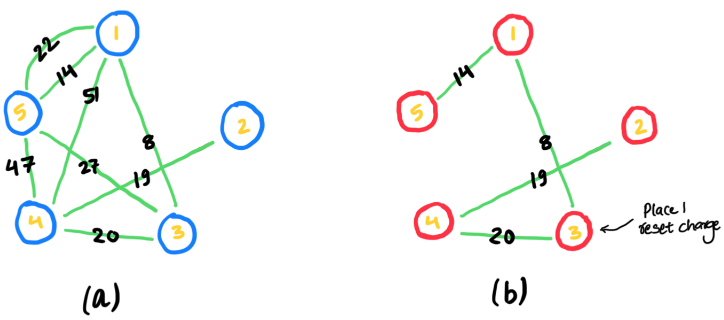

Figure 2: (a) Timeline 1 from sample case. (b) Placing 1 reset charge at any time-node can reset all the five time-nodes.

Marking Scheme (Tentative)

- ▷ **P1: Multiverse of Madnessss! .... 100 marks**
  - **Helping your Loki .... 20 marks**
    - ⋆ Implementation .... 20 marks
  - **Helping** Q **other Lokis .... 35 marks**
    - ⋆ Inefficient Implementation .... 10 marks
    - ⋆ Efficient Implementation .... 35 marks
  - **Basic Test Cases .... 20 marks**
  - **Large Test Cases .... 20 marks**
  - **Memory Deallocation, Coding Style, Indentation & Comments .... 5 marks**
- ▷ **Bonus: Resetting the timeline .... 30 marks**
  - Implementation .... 15 marks
  - Test Cases .... 10 marks
  - Memory Deallocation, Coding Style, Indentation & Comments .... 5 marks