Homework #3

Due Time: 2023/12/05 14:20
Contact TAs: `ada-ta@csie.ntu.edu.tw`

Instructions and Announcements

- There are **three programming problems** and **one hand-written problems. The maximum score for this assignment is 75 points.**
- **Programming.** The judging system is located at https://ada-judge.csie.ntu.edu.tw. Please log in and submit your code for the programming problems (i.e., those containing "Programming" in the problem title) by the deadline. NO LATE SUBMISSION IS ALLOWED.
- **Hand-written.** For other problems (also known as the "hand-written problems"), you should upload your answer to **Gradescope**. For each sub-problem, please label (on Gradescope) the corresponding pages where your work shows up, **or you may get some penalty for each problem you did not label** . NO LATE SUBMISSION IS ALLOWED.
- **Collaboration policy.** Discussions with others are strongly encouraged. However, you should write down your solutions **in your own words**. In addition, for **each and every** problem you have to specify the references (e.g., the Internet URL you consulted with or the people you discussed with, or **screenshot of the full conversation** if you are using GPT) on the **first** page of your solution to that problem and tag the corresponding page in Problem 0 on Gradescope. You may get zero points due to the lack of references.
- **Tips for programming problems.** Here are some C++ only tips. For this homework, you might want to check out sections about cin/cout, vectors, pairs, references, and range for. Reference is IMHO the most important since it rids you of C pointers. (Yeah!!!)
- **Instruction on explaining your algorithm with pseudocode.** If you decide to use pseudocode to describe your algorithm, please try to make your pseudocode human-readable (not C/C++ compilable), and your pseudocode should be less than 30 lines. Pseudocode without explanation or not human-readable may lead to a deduction of points for the corresponding problem. You can check out the textbook for some examples of easy-understanding pseudo code.

Problem 1 - 8e7's Race (Programming) (15 points)

Problem Description

When he's not thinking about the next ADA problem, 8e7 likes to jog around the NTU campus. On some occasions, he jogs with a very fast runner - Brian the Bullet Train. Because Brian the Bullet Train is much faster than 8e7, whenever he races 8e7, he could run a much longer route whilst still arriving sooner. 8e7 decides that it is his time to win for once, so he wants to find a shorter route for his run.

The NTU campus can be modeled as a graph with N junctions (vertices) and M bidirectional roads (edges), where each road has a positive length w_i. The junctions are labeled from 1 to N. The race starts on the CSIE department building, which is at the junction 1. In the race, both 8e7 and Brian the Bullet Train would start at the CSIE department building, visit some junctions, and arrive back at the department building. Note that they must visit at least one other junction that is not the department building.

In addition, they have agreed on a rule that they **must not pass through the same junction or the same road twice** in their respective runs (except for the department building at the start and end).

Because 8e7 wants to win the race, he tries to find the route that has the shortest length. The length of a route is the sum of lengths of roads visited in the route. Given the map of the campus, can you find the length of the shortest route that satisfies the requirements?

Input

On the first line, there are two integers N, M. The next M lines have three integers on each line u_i, v_i, w_i, meaning that there is a road from u_i to v_i that has length w_i.

Constraints:

- $3 \leq N, M \leq 3 \times 10^5$
- $1 \leq u_i, v_i \leq N$
- $1 \leq w_i \leq 10^9$
- No road connects a junction to itself. That is, $\forall i, u_i \neq v_i$.
- No two roads connect the same junctions. That is, $\forall i \neq j, (u_i, v_i) \neq (u_j, v_j), (u_i, v_i) \neq (v_j, u_j)$.
- It is possible to reach every point from junction 1 through some number of roads.
- It is guaranteed that there is at least one route that satisfies the requirements for 8e7.

Output

Output one line with an integer: the length of the shortest route that satisfies the requirements above. **The answer might need to be stored with long long integers.**

Test Group 0 (0 %)

- Sample input.

Test Group 1 (10 %)

- $N \leq 500$

Test Group 2 (30 %)

- There are at most 10 junctions directly connected to the CSIE department building.

Test Group 3 (60 %)

- No additional constraints.

Sample Input 1

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

Sample Output 1

```
7
```

Sample Input 2

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

Sample Output 2

```
9
```

Hint

- In this problem, you may view junctions as the vertices in the graph, and roads as the edges in the graph.
- For Sample Input 1, the optimal route for 8e7 is to visit junctions $[1, 3, 4, 1]$ in order, the total length is $1 + 2 + 4 = 7$. The route $[1, 2, 3, 1]$ is also a valid solution.
- Think about how you would solve the first two subtasks, and determine which types of shortest path algorithms are fit for them.
- Try drawing some cases and think about what properties the shortest route might have. In particular, it may be helpful to split the route into multiple sections and reason about them separately.

Problem 2 - XXLee's Bubble City (Programming) (15 points)

Problem Description

As a bubble tea enthusiast, XXLee's ambition was to build a city famous for delicious bubble tea and numerous bubble tea shops. After years of dedication to bubble tea research, he finally became the most influential bubble tea scholar in the world. Recently, he was elected as the mayor of Bubble City.

There are N bubble tea shops in Bubble City. To make sure that citizens and tourists can conveniently move between bubble tea shops, Bubble City has a special metro network, named Bubble Metro. Bubble Metro consists of M **one-way** roads, where each road directly connects two distinct bubble tea shops.

XXLee would like to announce a new travel plan to encourage more people to come to enjoy bubble tea in Bubble City. The travel plan includes a list of recommended bubble tea shops. XXLee hopes that the number of shops on the list is **as large as possible** and they should satisfy the condition: For any two bubble tea shops x, y on the list, one can move from x to y **or** move from y to x by Bubble Metro. Notice that it does **not** need to guarantee that one can move **both** from x to y and from y to x.

Given the roads in Bubble Metro, please tell XXLee the maximum number of shops on the list.

Input

The first line contains two positive integers N, M, representing the number of bubble tea shops in Bubble City and the number of roads in Bubble Metro.

There are M lines that follow. In the i-th line of them, there are two integers u_i, v_i, representing that the i-th road in Bubble Metro is from u_i to v_i.

- $1 \leq N, M \leq 10^6$
- $1 \leq u_i, v_i \leq N$
- $u_i \neq v_i$
- $\forall i \neq j, (u_i, v_i) \neq (u_j, v_j)$

Output

Output an integer representing the maximum possible number of shops on the list.

Test Group 0 (0 %)

- Sample input.

Test Group 1 (20 %)

- There is no cycle in Bubble Metro.[1]

Test Group 2 (20 %)

- $N, M \leq 5000$

Test Group 3 (60 %)

- No additional constraints.

[1] Formally, there doesn't exist a sequence of shops $p_1, p_2, \ldots, p_k$ s.t. for each $1 \leq i < k$, there is a road from p_i to p_{i+1} and there is a road from p_k to p_1.

Sample Input 1

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

Sample Output 1

```
7
```

Sample Input 2

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

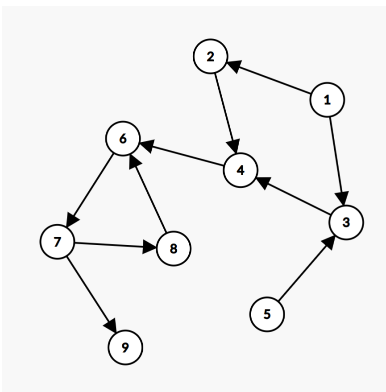
Sample Output 2

```
6
```

Hint

- You may view shops as the vertices of the graph, and one-way roads as the directed edges in the graph.
- **Test Group 1** might be helpful.

The figure shows the Bubble Network in Sample Input 1:



A list with the maximum number of shops is $\{1, 3, 4, 6, 7, 8, 9\}$.

Problem 3 - Alexander and Maze (Programming) (15 points)

Problem Description

Because Alexander was playing the Watermelon Game at 4 a.m., he missed his practice session with Billy. Billy was very annoyed by Alexander's absence, so he decided to play pranks on Alexander.

Billy constructed a maze with multiple rooms (i.e. vertices) and transport gates (i.e. edges). He told Alexander that if he could go through the maze within **8** minutes, then he would give Alexander 49 one-dollar coins. However, he did not tell Alexander that using a transport gate takes **8** hours, meaning Alexander would get stuck at the first transport gate. Being attracted to his precious one-dollar coins, Alexander agreed immediately without noticing Billy's evil plan.

After **8** hours, Alexander came out from the first transport gate in a rage. He asked Billy to remove the time constraint and create an additional transport gate to compensate for him. Upon listening to Alexander's request, another evil plan came to Billy's mind: adding a gate that would not lower the minimum number of transport gates needed to go from the source to the destination. Billy was very good at math, so he enumerated all the combinations that met the requirement.

TL; DR: You may view rooms as the vertices of the graph, and the transport gates as the directed edges of the graph. The problem is to find out the ways to add a single edge so that adding this edge would not lower the distance between source and destination.

Input

The first line contains four integers n, m, s, t representing the number of rooms, the number of transport gates, the source and the destination. Each of the next m lines contains two integers u, v which means that Alexander can go from u to v through a transport gate.

- $3 \leq n \leq 3 \times 10^5$
- $1 \leq m \leq \min(3 \times 10^5, n^2)$
- $1 \leq s, t, u_i, v_i \leq n$
- $s \neq t$
- $(u_i, v_i) \neq (u_j, v_j)$ if $i \neq j$

Output

Output the number of the combinations that met the requirement. Notice that you can not create a transport gate that already exist.

Test Group 0 (0 %)

- Sample Input.

Test Group 1 (10 %)

- The input forms a linked list, while s is the head and t is the tail.

Test Group 2 (20 %)

- The input forms a directed tree, while s is the root and t is a leaf.

Test Group 3 (70 %)

- No additional constraints.

Sample Input 1

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

Sample Output 1

```
5
```

Sample Input 2

```
3 1 1 3
1 3
```

Sample Output 2

```
8
```

Explanation

In sample 1 we can add $(1,1),(2,1),(2,2),(3,2),(3,3)$.

In sample 2 we can add everything except $(1,3)$.

Problem 4 - Graphics (35 points)

You should prove the correctness and time complexity for your algorithm in (a-2), (a-3), (b-1), and (b-2). The algorithms taught in class can be applied directly.

Pseudo-code is not permitted, and it is recommended that your answer should fit within 1 A4 page for each subproblem, using 12pt size text or readable handwritten text.

(a) Longest trip of Xian Chong Country (20 pts)

Xian Chong Country consists of n islands connected by $n-1$ bidirectional bridges (that is, it forms a **tree**). You can travel between two islands only if there is a bridge connecting them. It is guaranteed that you can reach any island from any other island by using a series of bridges.

If the path from island A to island B passes through k islands, and there are no islands C and D such that the path from island C to island D passes through more than k islands, then we define the path from A to B as the **longest trip**.

One day, Alexander and Arctan arrived in Xian Chong country for a visit, and they wanted to stay in Xian Chong country as long as possible without visiting the same island twice. Therefore, both of them took a trip according to the following rules:

- Both of them should take a **longest trip**. However, the route may be different for Alexander and Arctan.
- When arriving on an island, stay on the island for one day and proceed to the next island on the next day.

1. (5pts) Prove that Alexander and Arctan will meet each other. Formally, they will either arrive on the same island on the same day, or pass through the same bridge (could be in opposite directions) on the same day. In other words, prove that they will meet during the trip.
2. (5pts) Please design an algorithm to check which islands can be visited during one of the **longest trips** in Xian Chong Country.

   The algorithm's time complexity should be $O(n)$.
3. (10pts) Please design an algorithm to calculate the number of the **longest trips** in Xian Chong Country.

   The time complexity of the algorithm should be $O(n)$, and the path from A to B and the path from B to A is not the same.

(b) Grid King and Albert (15 pts)

Grid Kingdom is divided into $(m-1)\times(n-1)$ grids by m vertical roads and n horizontal roads. The king of Grid Kingdom, known as Grid King, is praised by the masses for his exceptional mathematical abilities. One of Grid King's fans, Albert, has come to Grid Kingdom to praise Grid King. However, Albert is actually coming to ridicule Grid King, since his red-black performance is better than Grid King's.

Initially, Albert is at the intersection in the bottom-left corner $(1,1)$, while Grid King is at the intersection in the top-right corner (m,n). Upon learning of Albert's visit to praise him, Grid King discreetly planted landmines at certain intersections to deter Albert's ridicule. These landmines are located at $(x_1, y_1), (x_2, y_2), \ldots, (x_c, y_c)$, excluding the intersections $(1, 1)$ and (m, n), where $0 \leq c \leq mn - 2$.

As a result, Albert has to avoid these landmines while moving only along the roads and walks from $(1, 1)$ to (m, n). Assuming that

- It takes a seconds to move from one intersection to an adjacent one.
- It takes b seconds to make a turn at an intersection.
- There are c landmines in the Grid.

1. (5pts) Please design an algorithm to calculate the shortest time it takes for Albert to walk from $(1, 1)$ to (m, n). You can use ∞ to represent that it is unreachable.

   The time complexity of the algorithm should be $O(mn(\log m + \log n))$, regardless of the value of a, b, c, since Albert doesn't like abc sneak in the time complexity.

2. (10pts) If Albert can remove at most d landmines before he moves, please design an algorithm to calculate the shortest time it takes for Albert to walk from $(1, 1)$ to (m, n) after removing landmines. You can use ∞ to represent that it is unreachable.

   The time complexity of the algorithm should be $O(mn(m + n)(\log m + \log n))$, regardless of the values of a, b, c and d.

Hint

- Construct the graph based on the problem description first.
- Convert the original problems to the problems taught in class.