Competitive Training Camp

Lecture 2

Christian Yongwhan Lim

Tuesday, July 15, 2025

Christian Yongwhan Lim

MIT Columbia University icpc foundation

Education

Stanford University

MIT

Part-time Jobs

Cornell Tech Jacobs Institute

MIT EECS

In lumine tuo videbimus lumen

Full-time Job

Google Research

TWO SIGMA

Workshops

Stanford ENGINEERING | Stanford Computer Forum

Carnegie Mellon University

Harvard

University of California

KAIST

NUS National University of Singapore

Coach/Judge

icpc International Collegiate Programming Contest

icpc.foundation

advancing the art and sport of competitive programming

https://www.yongwhan.io

Christian Yongwhan Lim

- **Head Coach**, Columbia ICPC Team
- **Chief Judge**, ICPC NA Mid-Central
- **Judge**, ICPC NA Qualifiers and Regionals
- **Chair**, ICPC CLI Symposium

https://www.yongwhan.io

Christian Yongwhan Lim

- **Head Coach**, Columbia ICPC Team
- **Chief Judge**, ICPC NA Mid-Central
- **Judge**, ICPC NA Qualifiers and Regionals
- **Chair**, ICPC CLI Symposium

- **Director**, ICPC Internships
- **Adjunct**, Columbia CS
- **VP of Engineering**, Arklex AI

https://www.yongwhan.io

Who are you?

- **Quick Introduction!**

Disjoint Set Union: Intuition

- `make_set(v)`
- `find_set(v)`
- `union_sets(a, b)`

- path compression
- union by size/rank

Disjoint Set Union: Implementation

```
void make_set(int v) {
  parent[v] = v;
  size[v] = 1;
}

int find_set(int v) {
  if (v == parent[v])
    return v;
  return find_set(parent[v]);
}
```

Disjoint Set Union: Implementation

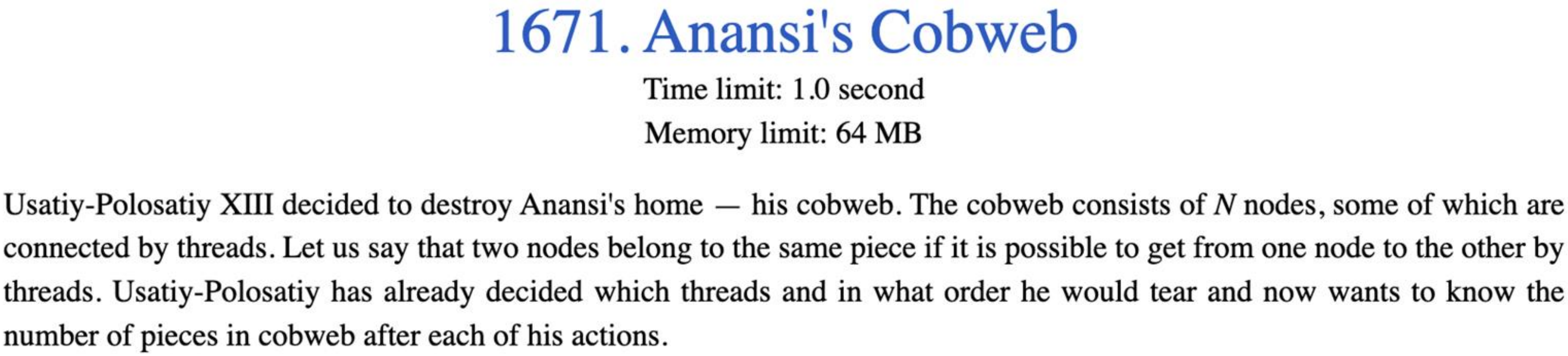
```
void union_sets(int a, int b) {
  a = find_set(a);
  b = find_set(b);
  if (a != b) {
    if (size[a] < size[b])
      swap(a, b);
    parent[b] = a;
    size[a] += size[b];
  }
}
```

Disjoint Set Union: Example

- Minimum Spanning Tree: **Kruskal's Algorithm**

Disjoint Set Union: Problem Statement

- https://acm.timus.ru/problem.aspx?space=1&num=1671



1671. Anansi's Cobweb

Time limit: 1.0 second

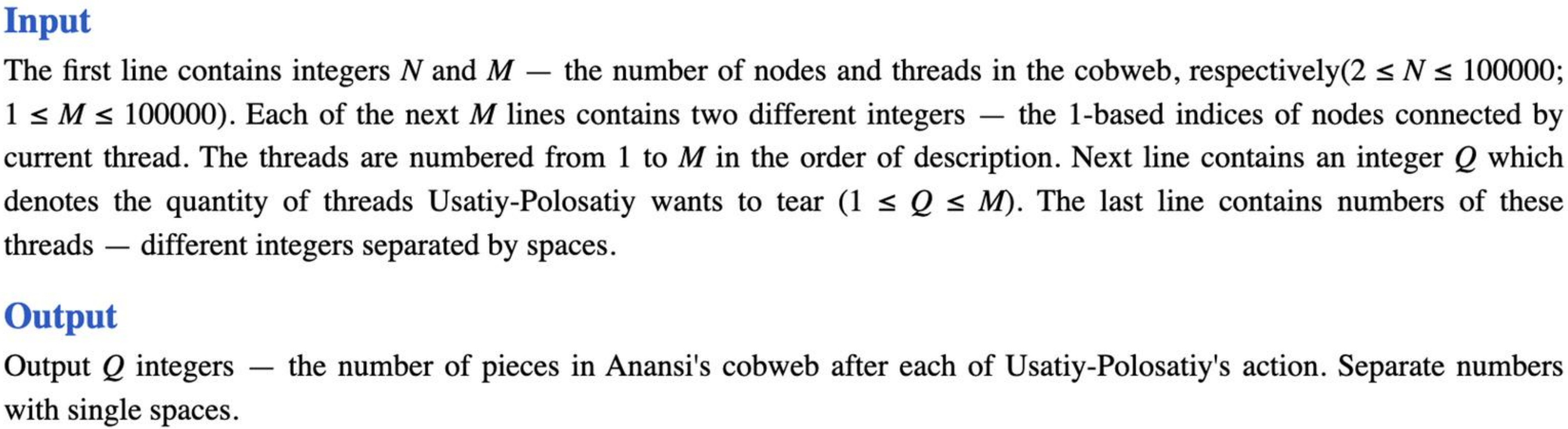
Memory limit: 64 MB

Usatiy-Polosatiy XIII decided to destroy Anansi's home — his cobweb. The cobweb consists of *N* nodes, some of which are connected by threads. Let us say that two nodes belong to the same piece if it is possible to get from one node to the other by threads. Usatiy-Polosatiy has already decided which threads and in what order he would tear and now wants to know the number of pieces in cobweb after each of his actions.

- Notice **time** and **memory (space)** limits! This is the usual pattern!

Disjoint Set Union: Input/Output Specification

- https://acm.timus.ru/problem.aspx?space=1&num=1671



Input

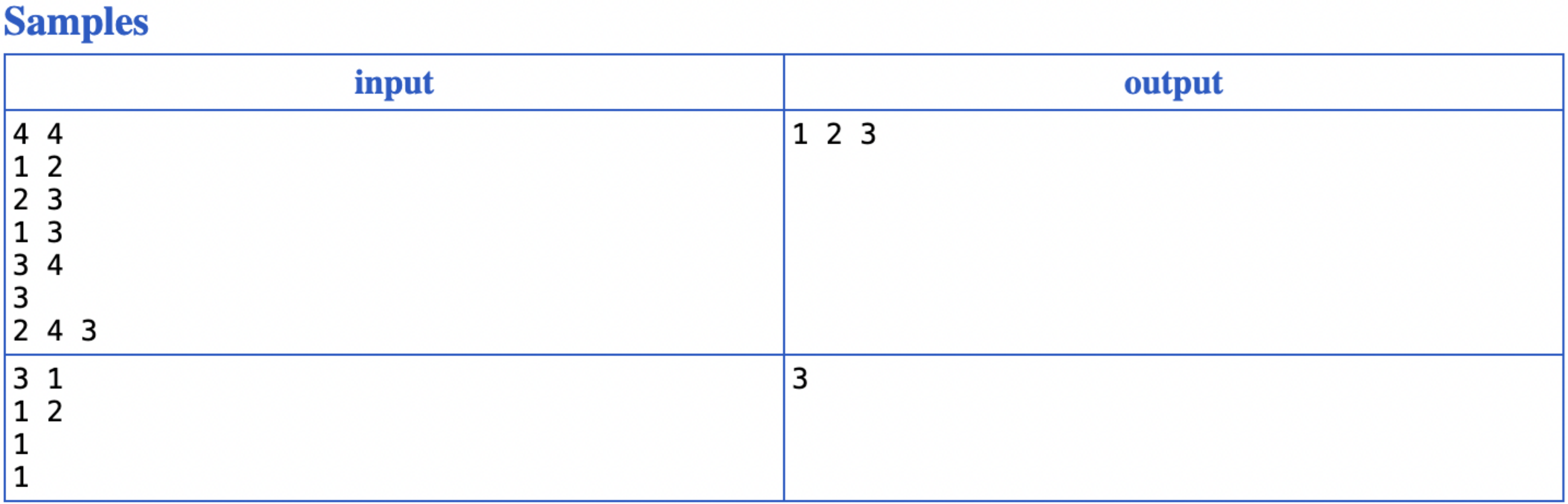
The first line contains integers N and M — the number of nodes and threads in the cobweb, respectively($2 \le N \le 100000$; $1 \le M \le 100000$). Each of the next M lines contains two different integers — the 1-based indices of nodes connected by current thread. The threads are numbered from 1 to M in the order of description. Next line contains an integer Q which denotes the quantity of threads Usatiy-Polosatiy wants to tear ($1 \le Q \le M$). The last line contains numbers of these threads — different integers separated by spaces.

Output

Output Q integers — the number of pieces in Anansi's cobweb after each of Usatiy-Polosatiy's action. Separate numbers with single spaces.

Disjoint Set Union: Sample

- https://acm.timus.ru/problem.aspx?space=1&num=1671



Samples

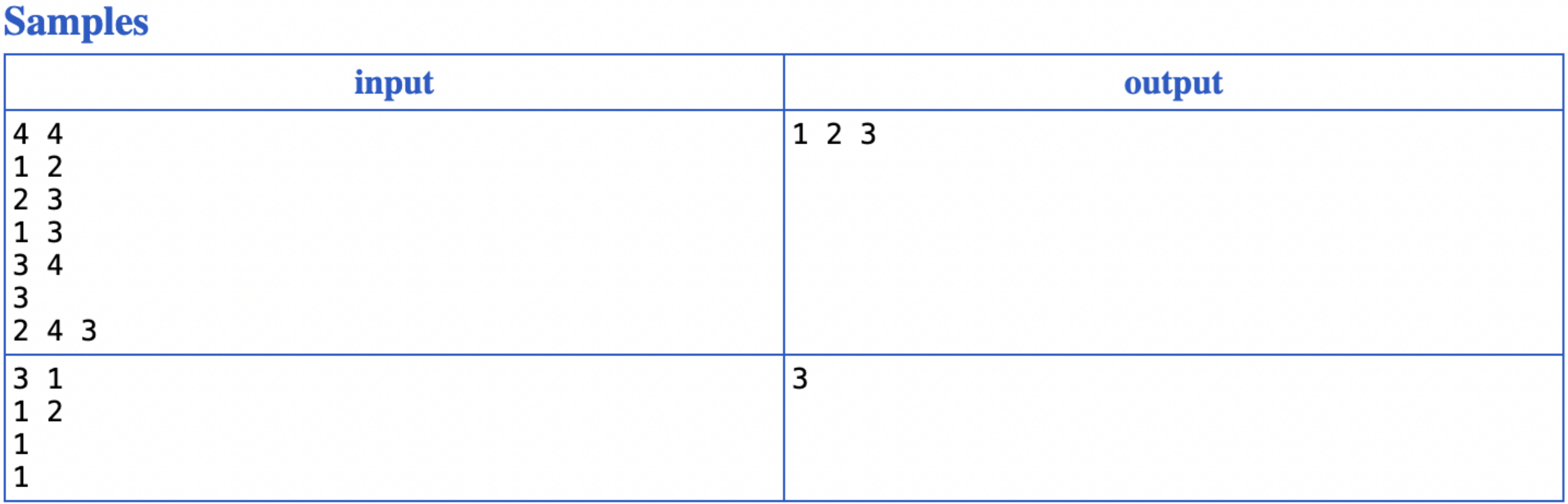
input	output

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

3 1
1 2
1
1
	3

Disjoint Set Union: Sample Explanation

- https://acm.timus.ru/problem.aspx?space=1&num=1671



Samples

input	output

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

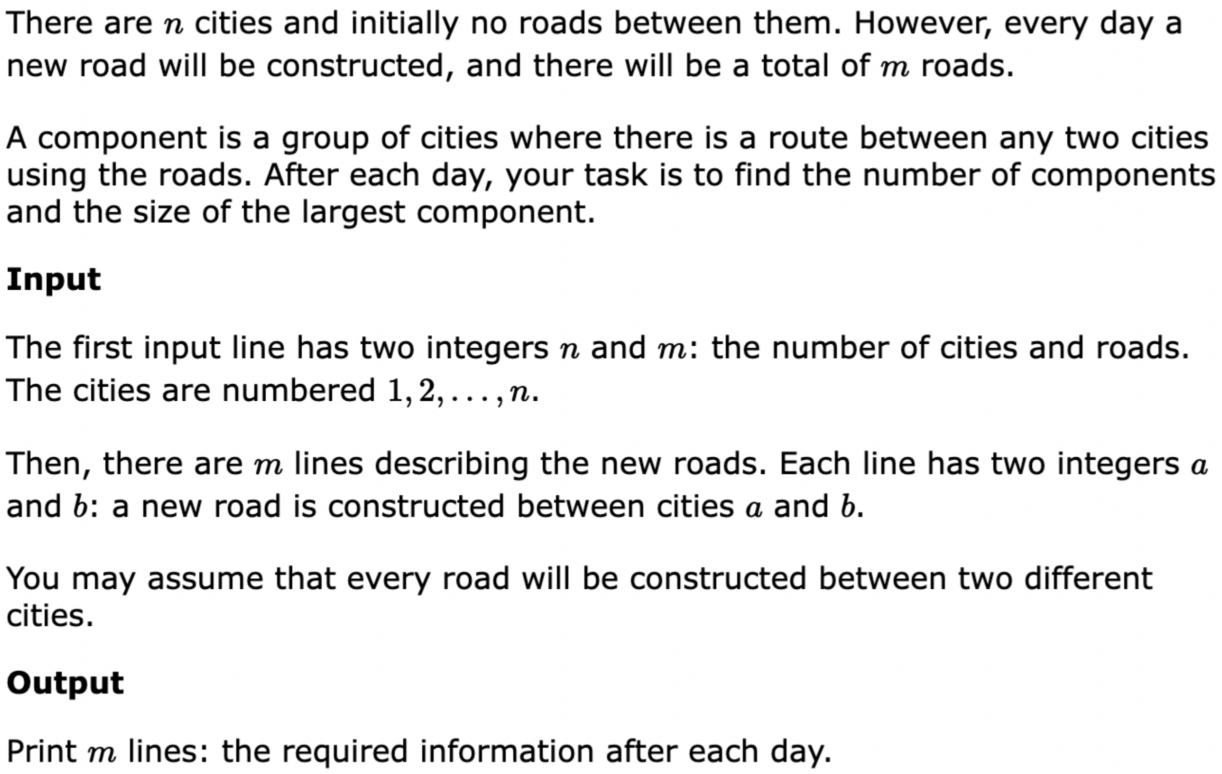
3 1
1 2
1
1
	3

Discuss for few minutes!

Solution Idea

- Disjoint Set Union, but apply them backwards;
- At the end, print the result after reversing;

Road Construction (CSES 1676)



There are n cities and initially no roads between them. However, every day a new road will be constructed, and there will be a total of m roads.

A component is a group of cities where there is a route between any two cities using the roads. After each day, your task is to find the number of components and the size of the largest component.

Input

The first input line has two integers n and m: the number of cities and roads. The cities are numbered $1, 2, \dots, n$.

Then, there are m lines describing the new roads. Each line has two integers a and b: a new road is constructed between cities a and b.

You may assume that every road will be constructed between two different cities.

Output

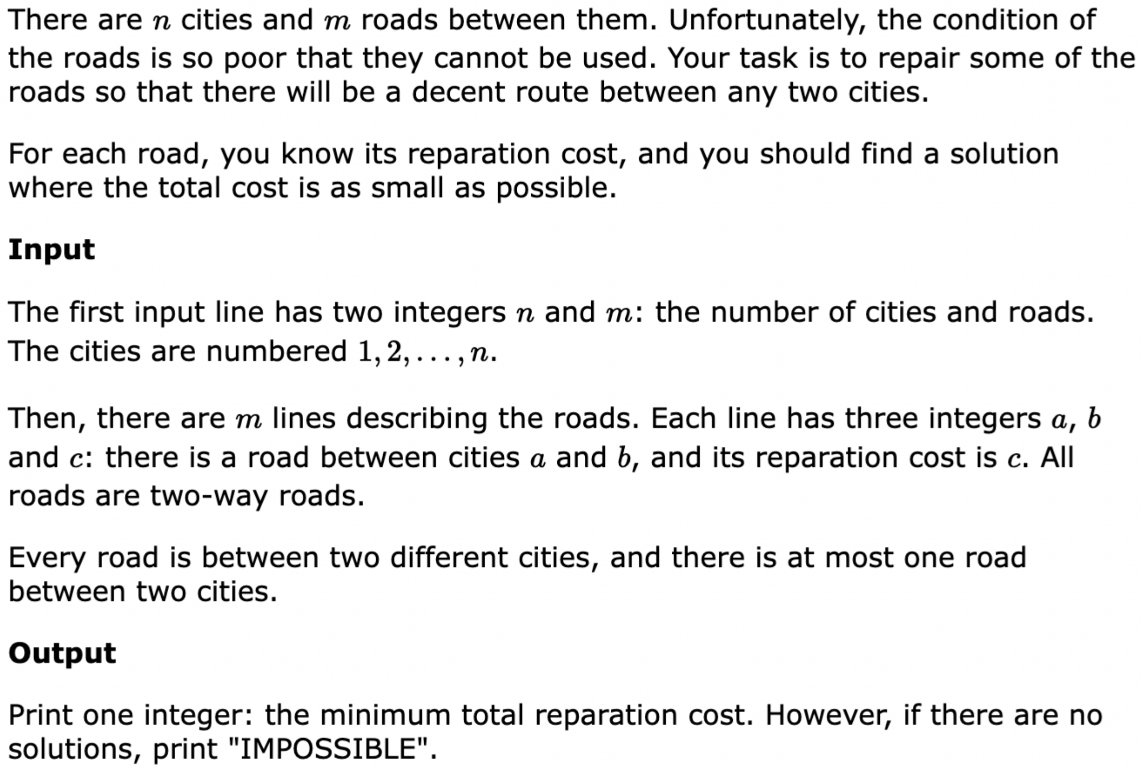
Print m lines: the required information after each day.

Discuss for few minutes!

Solution Idea?

- **Disjoint Set Union!**

Road Reparation (CSES 1675)



There are n cities and m roads between them. Unfortunately, the condition of the roads is so poor that they cannot be used. Your task is to repair some of the roads so that there will be a decent route between any two cities.

For each road, you know its reparation cost, and you should find a solution where the total cost is as small as possible.

Input

The first input line has two integers n and m: the number of cities and roads. The cities are numbered $1, 2, \dots, n$.

Then, there are m lines describing the roads. Each line has three integers a, b and c: there is a road between cities a and b, and its reparation cost is c. All roads are two-way roads.

Every road is between two different cities, and there is at most one road between two cities.

Output

Print one integer: the minimum total reparation cost. However, if there are no solutions, print "IMPOSSIBLE".

Discuss for few minutes!

Solution Idea?

- **Kruskal!**