COP 3503C Programming Assignment # 3

Disjoint Set

Read all the pages before starting to write your code

What should you submit?

Write all the code in a single Main.java file and submit it. Also submit **Lastname_testStrategy.txt**. The testing strategy file will briefly discuss how you have tested your code to make sure your code is really able to solve the problem. **Submit your** Main.java and the test strategy file **to Codegrade platform.**

Please include the following commented lines in the beginning of your code to declare your authorship of the code:

/*

Summer 24

COP 3503C Assignment 3

This program is written by: Your Full Name */

Compliance with Rules: UCF Golden rules apply towards this assignment and submission. Assignment rules mentioned in syllabus, are also applied in this submission. The TA and Instructor can call any students for explaining any part of the code in order to better assess your authorship and for further clarification if needed.

Caution!!!

Sharing this assignment description (fully or partly) as well as your code (fully or partly) to anyone/anywhere is a violation of the policy. I may report to office of student conduct and an investigation can easily trace the student who shared/posted it. Also, getting a part of code from anywhere will be considered as cheating.

Deadline:

See the deadline in webcourses. An assignment submitted by email will not be graded and such emails will not be replied according to the course policy.

What to do if you need clarification on the problem?

I will create a discussion thread in webcourses and I highly encourage you to ask your question in the discussion board. Maybe many students might have same question like you. Also, other students can reply and you might get your answer faster. Also, you can write an email to the TAs and put the course teacher in the cc for clarification on the requirements.

How to get help if you are stuck?

According to the course policy, all the helps should be taken during office hours. Occasionally, we might reply in email.

Problem Description: Destroy the connections

You have vowed to defeat your sworn enemy, but warfare in the 21st century isn't fought with guns. Knowledge is power, and knowledge is disseminated through computer networks. Thus, to defeat your sworn enemy, you plan on cutting the wired connections between pairs of computers to reduce the overall connectivity of your enemy's network.

We can model a computer network as pairs of connections between computers, also known as an undirected graph. We can define the connectivity of a network as the sum of the sizes squared of each of the separate components of this graph. For the example graph shown below, the current connectivity equals $2^2 + 6^2 + 1^2 = 41$.
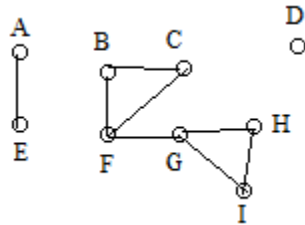
If you were to destroy the connection between computers F and G, then the new network would look like this
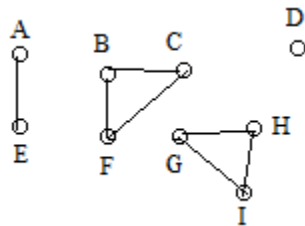
and its connectivity would only be $2^2 + 3^2 + 3^2 + 1^2 = 23$.

The Problem:

Given a network of n computers, a set of the pairs of computers that are initially connected, and a sequence of steps where connections are destroyed, one by one, calculate the connectivity (as defined in the problem specification above) after each connection is destroyed.

The Input (must be read from standard input (no file i/o. use of file i/o will fail all the test cases and you will get zero)):

The first line of input contains three space separated integers, n ($1 \le n \le 10^5$), m ($1 \le m \le 3 \times 10^5$), and d ($1 \le d \le m$), representing the number of computers in the enemy network, the number of connections between pairs of computers in the enemy network, and the number of those connections which you will destroy, respectively. The computers are numbered 1 through n, inclusive, and the connections are numbered 1 through m respectively.

The following m lines will each contain a pair of distinct integers, u and v ($1 \le u, v \le n, u \ne v$), representing that computers u and v are connected, initially. (Note: we denote the first of these lines as connection 1, the second as connection 2, and so forth.) It is guaranteed that each of the pairs listed will be unique pairs; namely, the same two values of u and v will never appear on two different lines as separate connections. (Of course, many individual computers will be connected to more than one other computer, so a particular value u may appear on more than one of these m lines.)

The following d lines will each contain a unique integer in between 1 and m, inclusive, representing a connection number that gets destroyed. These will appear in the order that they get destroyed.

The Output (standard console output):

Output $d+1$ lines of output. The first line should be the initial connectivity of the network. The following d lines should have the connectivity of the network after each connection is destroy, one by one.

Sample Input

9 8 2	
1 5	
2 3	
2 6	
3 6	
6 7	
7 8	
7 9	
8 9	
5 //it means 5th connection from the above list	
3	

Sample Output

```
41
23
23
```

3 3 3
1 2
1 3
2 3
3
1
2

```
9
9
5
3
```

Implementation Requirements

This assignment is testing the use of the disjoint set data structure. You must use a disjoint set to solve the problem even though other ways exist. You will have to modify the disjoint set shown in class to solve the given problem. The point of the assignment is to see if you can figure out how to do that modification on your own.

As always, your code should use good style, including but not limited to: a header comment, reasonable number of internal comments, good modular break down, good variable names, and utilizing objects and the Java API as appropriate for solving the problem,

What To Submit

For this assignment, please submit a single Java program named **`Main.java`** and **Lastname_testStrategy.txt** with `brief explanation of your testing strategy`

Hints:

1. The code will not be that straightforward, so you need to plan
2. Draw the sample input first and then see how the output numbers are calculated
3. You can use our existing disjoint set code and update it (make sure to use **path compression and union by rank approach**. Otherwise, your code may fail larger grading test cases due to optimization issue).
4. The union part of the cycle detection concept we have learned in the class could be useful

5. We mainly use the find and union operations of disjoint sets. As there is not an easy way to break a set when you remove an edge, it would be best idea not to think in that direction
6. **Instead, generate the result bottom up.**
    a. Make sets based on number of nodes
    b. Store all the edges in your data structure (maybe a class or 2d array). Maintaining status (exist/not exist) for each edge might be useful
        i. Don't use UNION operation during this process. If you do, you will get stuck as you will not be able to easily remove them
    c. Keep the list of index of the edges you are deleting and maintain proper edge status
    d. Also, have an array to store the result for each deletion
    e. For each deletion index, perform UNION and calculate result
    f. In this way, you will generate your result array and just print it
7. Also note that during this process, you will need to maintain the size of a set at the root of each set.
8. Of-course I did not give you the exact details and steps for the solution. However, the above keywords and logics should help you to think in the proper direction.

Good Luck!