CS29003 ALGORITHMS LABORATORY
ASSIGNMENT 8
Date: 21^{st} Oct, 2021

Important Instructions

1. **Input:** To be taken from command line
2. **Format fo files to be submitted to Moodle and HackerRank:** ROLLNO_A8.c/.cpp
3. Take inputs from command line in the specified format
4. You are to **stick to the file input output formats strictly** as per the instructions.
5. Submission through **.zip files are not allowed**.
6. Write your name and roll number at the beginning of your program.
7. Do not use any global variable unless you are explicitly instructed so.
8. Use proper indentation in your code.
9. Please follow all the guidelines. Failing to do so will cause you to lose marks.
10. **There will be part marking.**

Disjoint Sets

Problem Statement

Today, Arthur Dent is hitchhiking aboard a spaceship, *Vogon 101* that is travelling across the Universe. On another spaceship, *Heart of Gold*, is Mr. Zaphod Beeblebrox, the President of the Milky Way Galaxy. They are visiting different galaxies in their quest to find the answer to the "Ultimate Question to Life, the Universe, and Everything". Every galaxy that they visit has multiple planets and there are inter-planetary routes connecting these planets. These inter-planetary routes have been created to serve different kinds of travellers- hitchhikers, politicians, tourists, etc. Thus, there are some routes that only Mr. Zaphod can use because of his position as the President of the Milky Way Galaxy and some routes only Arthur can use, as a hitchhiker, along with common routes that both of them can use. One such interplanetary routing system can be seen in Figure 1, where the routes that Mr. Zaphod can use are colored in red, the routes that Arthur can use are colored in green and the routes that both of them are allowed to use are colored in blue. Similar color-coded paths exist in every galaxy that they visit today.

The most powerful supercomputer in the Universe, *Deep Thought* was tasked with identifying the number of direct edges that can be removed from this interplanetary routing system so that both of them have access to all the planets without having to traverse extra paths as the number of planets to cover is large and time is less. Unfortunately the supercomputer is busy finding the answer to other questions and so you have to do its job.

You can think of the galaxy as a graph with planets as the nodes and routes as the edges. You have to return the count of the **maximum** number of edges that you can remove from a given galaxy such that both Arthur and Zaphod can visit all planets within the galaxy, given that they start at some arbitrary planet. If no such solution exists, you must inform them of the same. Thankfully, you have recently studied about the **disjoint set** data structure and you can use that knowledge to solve the given problem.

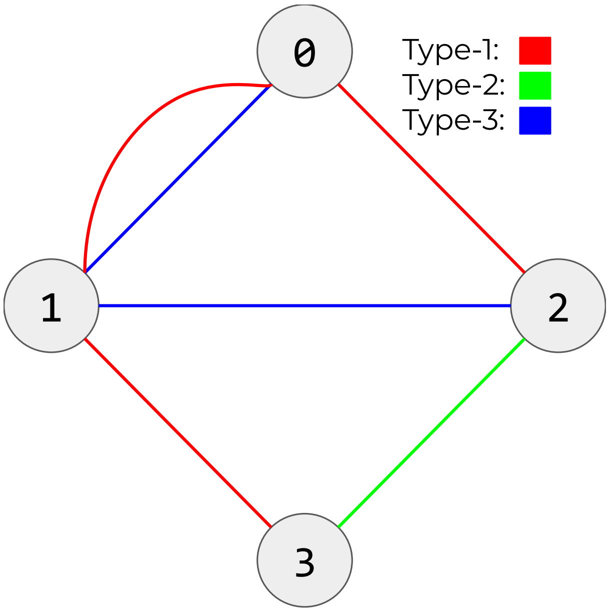

Figure 1: An example galaxy with the different planets and types of interplanetary routes

Part 1: Taking Input and Creating the Graph(30 marks)

The graph for each galaxy will be given as an input along with the type of each edge. There can be three types of edges, enumerated from 1 to 3-

▷ Type-1 : Edges that can be traversed only by Zaphod

▷ Type-2 : Edges that can be traversed only by Arthur

▷ Type-3 : Edges that can be traversed by both Zaphod and Arthur.

You must take as input the number of nodes in the graph, N, the number of edges, M and the edge list with the type of each edge, in the form of an edge list and print the resulting graph to the console in the form of a 2-D Matrix.

Input Format

▷ The first line is two space separated integers N and M, indicating the number of vertices and edges, respectively.

▷ The next M lines are the edges of the graph of the form $\langle T, u, v \rangle$ indicating the type of the edge, the source vertex and the sink vertex respectively. The **edges are bidirectional** so an edge $\langle u, v \rangle$ also implies the existence of the edge $\langle v, u \rangle$.

Output Format

For the given input, you must print the resulting graph in the form of a 2-D Matrix with the cell values indicating the type of edge that exists between the two pairs of vertices. If there exists more than 1 type of edge between a given pair of vertices, print them in appended format with the "#" delimiter as follows- t1#t2#t3 (where $t1 < t2 < t3$).

Write the following functions to store the graph and print the resulting 2-D matrix as explained above.

- ▷ **printGraph**: This should store the edge lists taken as input and then print the resulting graph as shown in the Sample Scenario below.

Part 2 : Counting number of edges to remove(70 marks)

After storing the graph in your intended format and printing the resultant 2-D matrix, you must now identify the maximum number of edges (of any type) that you must remove such that both Arthur and Zaphod should be able to traverse the entire galaxy using their allowed routes. You must return the count and if no solution exists, you must return -1.

Write the following functions for implementing the disjoint set data structure with **Path Compression** and **Union-by-Rank**.

- ▷ **maxEdgesToRemove**: This will take as input the input graph along with the types of all edges and output the maximum number of edges to remove.
- ▷ **find**: Given a node, this will find the root of that node and return it. Apply path compression while finding the root.
- ▷ **unionNode**: Given two distinct nodes, this will make the root pointer of one node point to the root of the other node. If both roots have the same rank, increase rank of new root by one.

Sample Scenarios

Example-1 The first test case presents the input format for the graph in Figure 1. There are 4 vertices and 6 edges. The first column from line 2 onwards indicates the edge type and the 2nd and 3rd values indicate the source and sink. All values are space separated.

FILE: *input-1.txt*

```
4 6
1 0 1
3 0 1
1 0 2
1 3 1
3 1 2
2 2 3
```

The output for part 1 should look like the following 2-D matrix. Each cell value is separated from its row neighbour by a tab(or 4 spaces). As informed earlier, if there is more than 1 type of edge between a pair of vertices, append the type IDs using the “#” delimiter as can be seen for the edges between vertices 0 and 1. This should be followed by a single empty line.

After this, you should output the solution for part 2 in the exact format as shown with the text “Edges Removed = ” followed by either the maximum number of edges that should be removed or -1.

FILE: *output-1.txt*

```
0    1#3    1    0
1#3    0    3    1
1    3    0    2
0    1    2    0

Edges Removed = 2
```

Explanation From Figure 1, it can be seen that if the two edges- $\langle 1, 0, 1\rangle$ and $\langle 1, 0, 2\rangle$ are removed, both Zaphod and Arthur can still traverse the full graph. Thus the maximum number of edges to remove is 2. Removing any more edges will make the graph disconnected for either or both of them.

Example-2 There are 4 vertices and 3 edges. The first column from line 2 onwards indicates the edge type and the 2nd and 3rd values indicate the source and sink. All values are space separated.

FILE: *input-2.txt*

```
4 3
2 2 3
1 0 1
3 1 2
```

The output for part 1 should look like the following 2-D matrix. Each cell value is separated from its row neighbour by a tab(or 4 spaces). As informed earlier, if there is more than 1 type of edge between a pair of vertices, append the type IDs using the "#" delimiter as can be seen for the edges between vertices 0 and 1. This should be followed by a single empty line.

After this, you should output the solution for part 2 in the exact format as shown with the text "Edges Removed = " followed by either the maximum number of edges that should be removed or -1.

FILE: *output-2.txt*

```
0    1    0    0
1    0    3    0
0    3    0    2
0    0    2    0

Edges Removed = -1
```

Explanation Since no edge removal will allow either Zaphod or Arthur to traverse the full graph, the answer has to be -1.

Example-3 There are 5 vertices and 6 edges. The first column from line 2 onwards indicates the edge type and the 2nd and 3rd values indicate the source and sink. All values are space separated.

FILE: *input-3.txt*

```
5 6
3 2 3
1 0 1
3 1 2
3 4 3
2 1 0
3 1 4
```

The output for part 1 should look like the following 2-D matrix. Each cell value is separated from its row neighbour by a tab(or 4 spaces). As informed earlier, if there is more than 1 type of edge between a pair of vertices, append the type IDs using the “#” delimiter as can be seen for the edges between vertices 0 and 1. This should be followed by a single empty line.

After this, you should output the solution for part 2 in the exact format as shown with the text “Edges Removed = ” followed by either the maximum number of edges that should be removed or -1.

FILE: *output-3.txt*

```
0    1#2    0   0    0
1#2    0    3    0    3
0    3    0    3    0
0    0    3    0    3
0    3    0    3    0
Edges Removed = 1
```

Explanation it can be seen that if any one of the edges out of- $\langle 3,1,2\rangle$, $\langle 3,2,3\rangle$, $\langle 3,4,3\rangle$ or $\langle 3,1,4\rangle$ are removed, both Zaphod and Arthur can still traverse the full graph. Thus the maximum number of edges to remove is 1. Removing any more edges will make the graph disconnected for either or both of them.