

CSE 4304-Data Structures Lab. Winter 23-24**Batch:** CSE 22**Date:** February 19, 2025**Target Group:** All**Topic:** Disjoint Set**Instructions:**

- Regardless of how you finish the lab tasks, you must submit the solutions in Google Classroom. In case I forget to upload the tasks there, CR should contact me. The deadline will always be 11:59 PM on the day the lab took place.
- Task naming format: fullID_T01L01_2A.c/cpp
- If you find any issues in the problem description/test cases, comment in the Google Classroom.
- If you find any tricky test cases that I didn't include but that others might forget to handle, please comment! I'll be happy to add them.
- Use appropriate comments in your code. This will help you recall the solution easily in the future.
- Obtained marks will vary based on the efficiency of the solution.
- Do not use <bits/stdc++.h> library.
- Modified sections will be marked with **BLUE** color.
- You can use the STL stack unless it's specifically mentioned that you should use manual functions.

Group	Tasks
2A	1 2 3 4
1B	1 2 3 4
1A	1 2 3 4
2B	1 2 3 4
Assignments	2A/1B: 1A/2B:

Task 1: Basic operations of DSU

Consider that N items are stored in a Disjoint Set data structure, identified as 0, 1, 2, ..., N-1. There can be three types of operation conducted on the items:

1. **initialization**: creates separate sets with each item and points itself as the representative.
2. **find(item)**: returns the set's representative containing that item.
3. **union(i,j)**: connects the representatives of two sets and forms a new set by adding the second representative with the first one (doesn't consider rank)

After each input, print the state of the 'representative array', where every i-th item contains its representative information.

Sample Input	Sample Output
10	
1	0 1 2 3 4 5 6 7 8 9
3 1 3	0 1 2 1 4 5 6 7 8 9
2 1	1
2 3	1
3 3 5	0 1 2 1 4 1 6 7 8 9
3 5 7	0 1 2 1 4 1 6 1 8 9
3 6 8	0 1 2 1 4 1 6 1 6 9
3 8 9	0 1 2 1 4 1 6 1 6 6
3 4 8	0 1 2 1 4 1 4 1 6 6
2 8	4
2 6	4
2 4	4
3 5 6	0 1 2 1 1 1 4 1 6 6
2 8	1

Task 2:

Improve the Union(i,j) function of Task1 by introducing the 'Union by Rank' approach, where i's representative will be connected with j's representative if it has a lesser or equal rank. Print the rank of each node after each union operation. Besides, optimize the Find function using the 'path compression technique'.

(Introduce a new display feature which will be initiated if Option 4 is given)

Sample Input	Sample Output
10	
1	0(0) 1(0) 2(0) 3(0) 4(0) 5(0) 6(0) 7(0) 8(0) 9(0)
3 1 3	0(0) 3(0) 2(0) 3(1) 4(0) 5(0) 6(0) 7(0) 8(0) 9(0)
2 1	3
2 3	3
3 3 5	0(0) 3(0) 2(0) 3(1) 4(0) 3(0) 6(0) 7(0) 8(0) 9(0)
3 5 7	0(0) 3(0) 2(0) 3(1) 4(0) 3(0) 6(0) 3(0) 8(0) 9(0)
3 6 8	0(0) 3(0) 2(0) 3(1) 4(0) 3(0) 8(0) 3(0) 8(1) 9(0)
3 8 9	0(0) 3(0) 2(0) 3(1) 4(0) 3(0) 8(0) 3(0) 8(1) 8(0)
3 4 8	0(0) 3(0) 2(0) 3(1) 8(0) 3(0) 8(0) 3(0) 8(1) 8(0)
2 8	8
2 6	8
2 4	8
3 5 6	0(0) 3(0) 2(0) 8(1) 8(0) 3(0) 8(0) 3(0) 8(2) 8(0)
2 8	8
2 3	8
2 7	8 (the path will get compressed in this find call)
4	0(0) 3(0) 2(0) 8(1) 8(0) 3(0) 8(0) 8(0) 8(2) 8(0)

Task 3

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.

Sample Input	Sample Output	Explanation
5 3 1 2 1 3 4 5	4 2 3 3 2 3	{1,2} {3} {4} {5} {1,2,3} {4} {5} {1,2,3} {4,5}
7 10 1 4 5 6 4 7 1 7 3 5 1 2 2 7 3 6 2 5 5 4	6 2 5 2 4 3 4 3 3 3 2 4 2 4 2 4 1 7 1 7	{1,4} {2} {3} {5} {6} {7} {1,4} {2} {3} {5,6} {7} {1,4,7} {2} {3} {5,6} No change {1,4,7} {2} {3,5,6} {1,2,4,7} {3,5,6} No change No change {1,2,3,4,5,6,7} No change

Task 4

The Organization of Incredible Clowns wants to improve relations between its member countries. There are n countries, and currently, exactly $n - 1$ roads connecting them. These roads allow every country to be reachable from any other. However, due to financial issues, the organization can only replace roads instead of building new ones freely. Every day, they can close one existing road and immediately build a new one in a different location.

Your task is to determine how many days it will take to rebuild the roads so that every country remains connected, and to draw a plan of closure of old roads and building of new ones.

Sample Input	Sample Output
2 1 2	0
7 1 2 2 3 3 1 4 5 5 6 6 7	1 3 1 3 7
9 1 2 2 3 3 4 4 5 5 2 6 7 1 7 7 8	1 5 2 7 9