CS 2420 Program 7

Union/Find

Implement a Union/Find using smart union (by size) and path compression.

Use Union/Find to solve the following problem.

A political candidate wants to determine the size of each of two factions of his party. Every voter has been given a unique id. The newspapers indicate who is attacking who, but haven’t determine which faction the candidates belong to. Every instance of an “attack” is assumed to be between members of different factions.

Determine the maximum size of any faction.

**Input Format**  
  
Each case contains an integer **N** denoting the number of number of voters. Each of the next lines will contain two different integers **A**,**B** denoting there was an attack by person **A** of person **B**. If **A** and **B** are assumed to be in the same group, ignore that attack. Note, you may not be able to identify all members of each group.

Input case1.txt

5

1 2

2 3

2 4

2 5

3 4

Output

Attack 1 2

Attack 2 3

Attack 2 4

Attack 2 5

Attack 3 4

Ignored Attack 3 4

Group 2 has 1 members

Group 3 has 4 members

case1.txt max Largest group is of size 4

Input case2.txt

10

2 6

1 5

3 4

3 9

7 9

5 8

4 10

Output

10

Attack 2 6

Attack 1 5

Attack 3 4

Attack 3 9

Attack 7 9

Attack 5 8

Attack 4 10

Group 2 has 1 members

Group 5 has 1 members

Group 6 has 1 members

Group 7 has 3 members

Group 8 has 2 members

Group 9 has 2 members

case2.txt max Largest group is of size 3

**Hints:**

The Union Find data structure should be in a separate file which is tested independently of the program requirements. The main program of your rivals class should be short and consist of calls to the methods which do the work.

When two members are involved in an attack, each member may want to remember the id of the attacker/attacked. Thus, when 2,5 are involved in an attack, any previous opponent of 2 is unioned with 5 and any previous opponent of 5 is unioned with 2.