Gebze Technical University
Computer Engineering

CSE 222 - 2019 Spring

HOMEWORK 8 REPORT

NEVRA GÜRSES
161044071

Ayşe Şerbetçi Turan

1 INTRODUCTION

1.1 Problem Definition

There is some ordered people pairs such that Person1,Person2.There is a popularity relation between this people.The relation is that, Person1 thinks Person2 is popular.This relation is also transitive.For example, if there is a relation Person1,Person2 and Person2,Person3 by ordered people pairs,there is also a relation as Person1,Person3 although it is not specified by the input pairs.Our task is to find number of people who are considered popular by every other person.We have to read txt file where is popularity relations in file as integer and first line is number of people and number of relation .And then we have to find number of popular people who are considered popular by every other person.

1.2 System Requirements

This program is writed with Java programming language.And Java requires following system requirements for Windows:

→Windows 10 (7u85 and above)
→Windows 8.x (Desktop)
→Windows 7 SP1.
→ Vista SP2.
→Windows Server 2008 SP2 and 2008 R2 SP1 (64-bit)
→Windows Server 2012 (64-bit) and 2012 R2 (64-bit)
→RAM: 128 MB; 64 MB for Windows XP (32-bit)
→Disk space: 124 MB.

This program is writed in Intellij IDEA.And Intellij IDEA requires following system requiremets:

→Windows 10/8/7/Vista/XP (incl. 64-bit).
→2 GB RAM minimum, 4 GB RAM recommended.
→1.5 GB hard disk space + at least 1 GB for caches.
→1024x768 minimum screen resolution.

This program also require Java SE Development Kit 11.0.2 .Java SE Development Kit (JDK) and Java SE Runtime Environment (JRE) require at minimum a Pentium 2 266 MHz processor.For the JDK, the option of installing the following features:

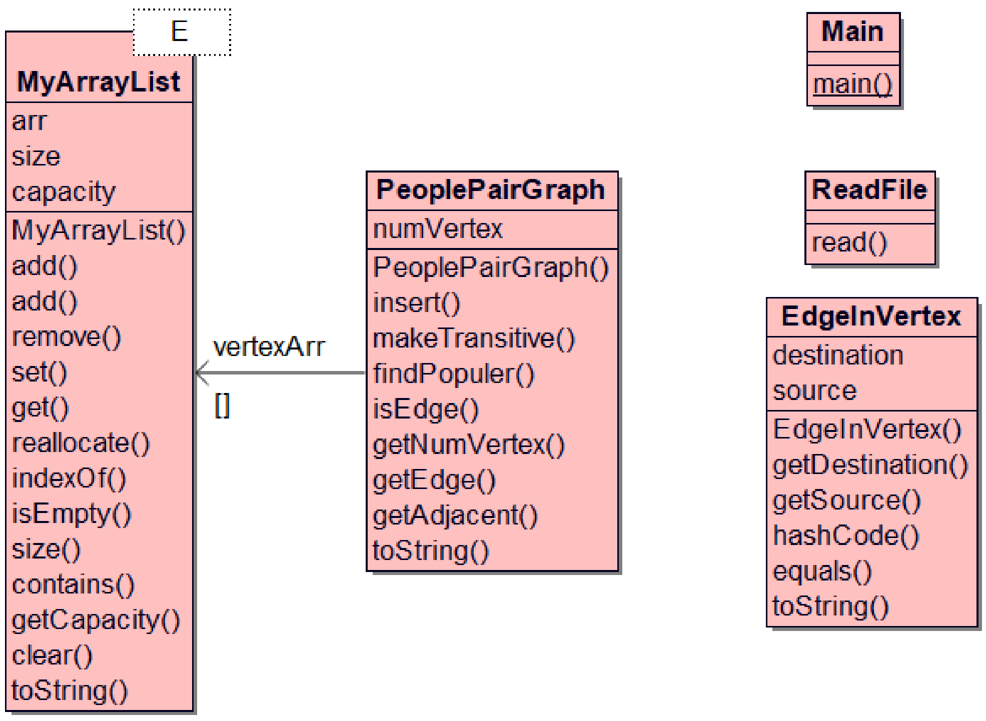
→Development Tools.

→Source Code.

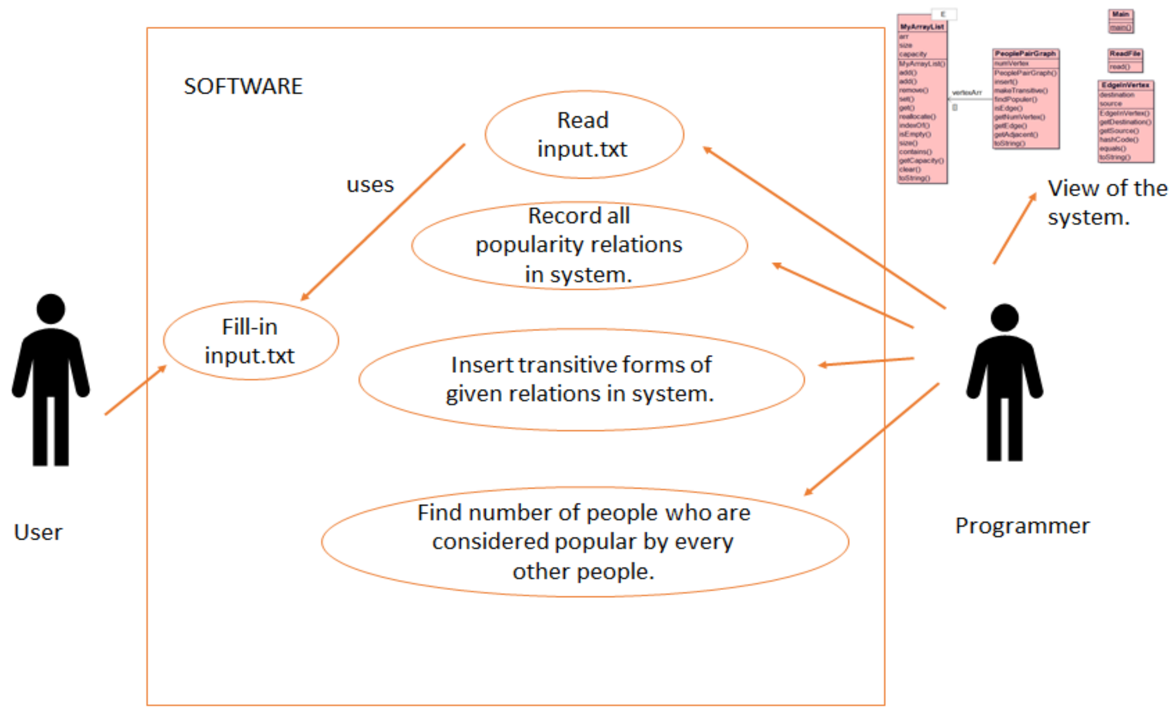
→Public Java Runtime Environment.

2 METHOD

2.1 Class Diagrams



2.2 Use Case Diagrams



2.3 Problem Solution Approach

This problem actually include a directed graph.Because the relations between people creates a directed graph relation.For example Person1 thinks that Person2 is popular, so Person1->Person2 .Actually Person2 is adjacent of Person1 according to graph structure.So I used graph data structure to solve this problem.I used adjacency list to create graph structure.I read txt file and I take number of people in constructor and then I create an EdgeInVertex class type ArrayList array with given number of people.The EdgeInVertex class is to hold source and destination point of an edge and to override equals and hashCode method.ArrayList array includes popularity relations.For example index 1 includes relations with P1 and people that P1 thinks popular.If index 1 include 1 as source point of an edge and 2 as destination point of an edge, this means that P1 thinks P2 is popular.I inserted ArrayList array all popularity relations in given input file.To make this relations transitive,after inserted all given popularity relations in ArrayList array,I create a makeTransitive() method to find all transitive relations.After finding all transitive relations , I find number of people who are considered popular by every other person by traversing

ArrayList array and controling whether current person is considered popular every other person except their own or not.If it is considered popular every other person , I increase counter.And so ,I control every person and calculate number of popular people.

3 RESULT

3.1 Test Cases

I test insert() method of my PeoplePairGraph class.I give several relations and my insert method adds relations as correctly.After that I tested makeTransitive() method and this method process correctly,it makes all given relations as transitive. After I read input file and I inserted all relations in graph,I printed all given relations and finded transitive relations of given relations in txt file in screen.All relations is true.And consequently,I printed calculated number of popular people who are considered popular by every other person.I write numbers that are in our homework pdf in input txt file and the output is same.And I tested many different relations and popularity calculations and the results are true.

3.2 Time Complexity Calculations:

PeoplePairGraph Class:

1)public void insert(int source,int destination) Method: This method has O(n) time complexity.Because in this method, contains method of arraylist class is used and this method has O(n) time complexity.So add method also has O(n) time complexity.

2) public void makeTransitive() Method: This method has O(n^3) time complexity.N is number of people.This method has cubic time complexity because there is inner three while loop each of has n time.So result is (n*n*n)=n^3.

3)public int findPopuler() Method: This method has O(n^2) time complexity.
In this method,there is two while loop.Each of has n time.So result is (n*n)=n^2.

4) public int getNumVertex(): This method has O(1) time complexity.There is only returned number of people in this method,so getNumVertex method has constant time complexity.

5) public boolean isEdge(int source, int dest): This method has O(n) time complexity. Because in this method, contains method of arraylist class is used and this method has O(n) time complexity.So add method also has O(n) time complexity.

6) public MyArrayList<Integer> getAdjacent(int source): This method has O(n) time complexity.Because all people are traversed in this method.

7)public EdgeInVertex getEdge(int source,int destination): method has O(n) time complexity because it includes contains method of arraylist class and this method has O(n) time complexity.So isEdge method also has O(n) time complexity.

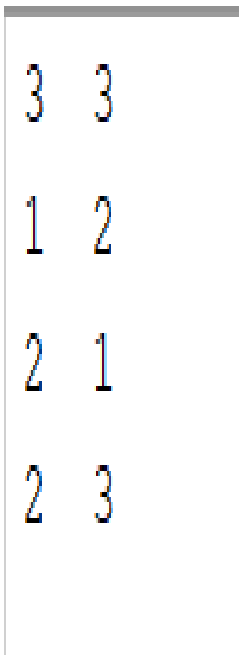
EdgeInVertex Class

All methods of this class is O(1) time complexity.Because there is doing only assignment or basic operations in this class.

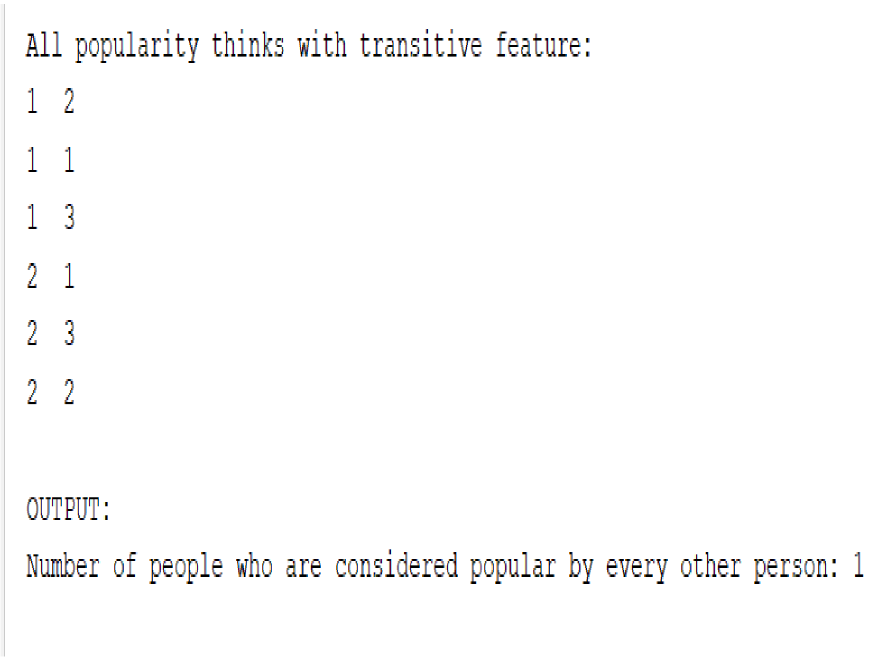
3.3 Running Results

Outputs with given ordered relations with transitive forms:

INPUT: OUTPUT:

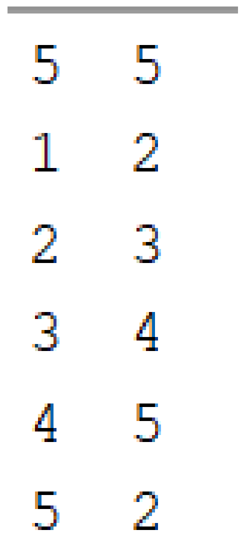


```
3 3
1 2
2 1
2 3
```



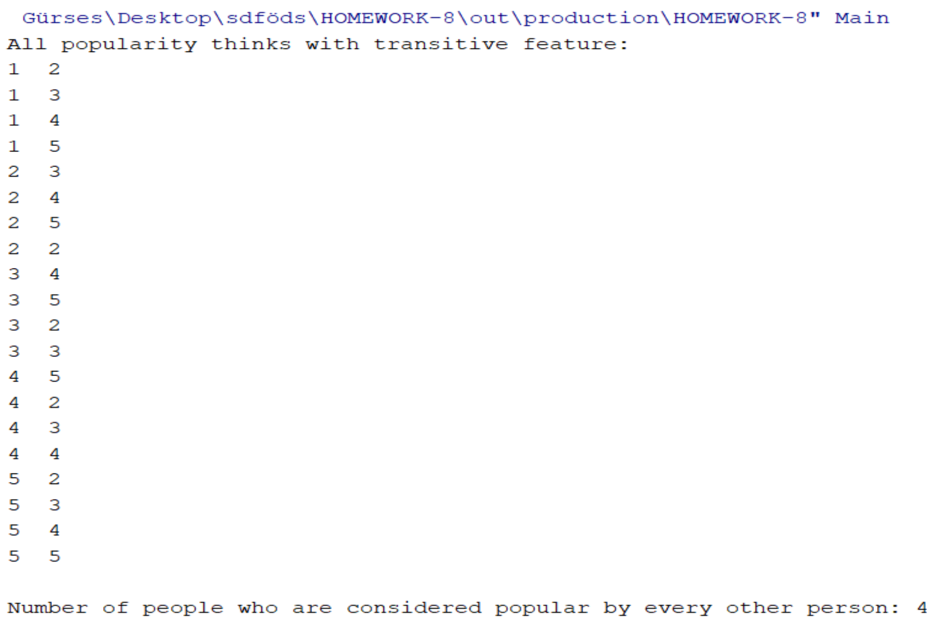
```
All popularity thinks with transitive feature:
1 2
1 1
1 3
2 1
2 3
2 2

OUTPUT:
Number of people who are considered popular by every other person: 1
```

INPUT:



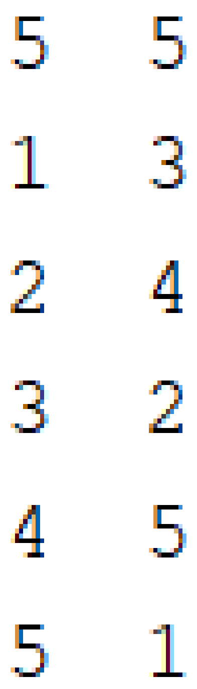
```
5 5
1 2
2 3
3 4
4 5
5 2
```

OUTPUT:



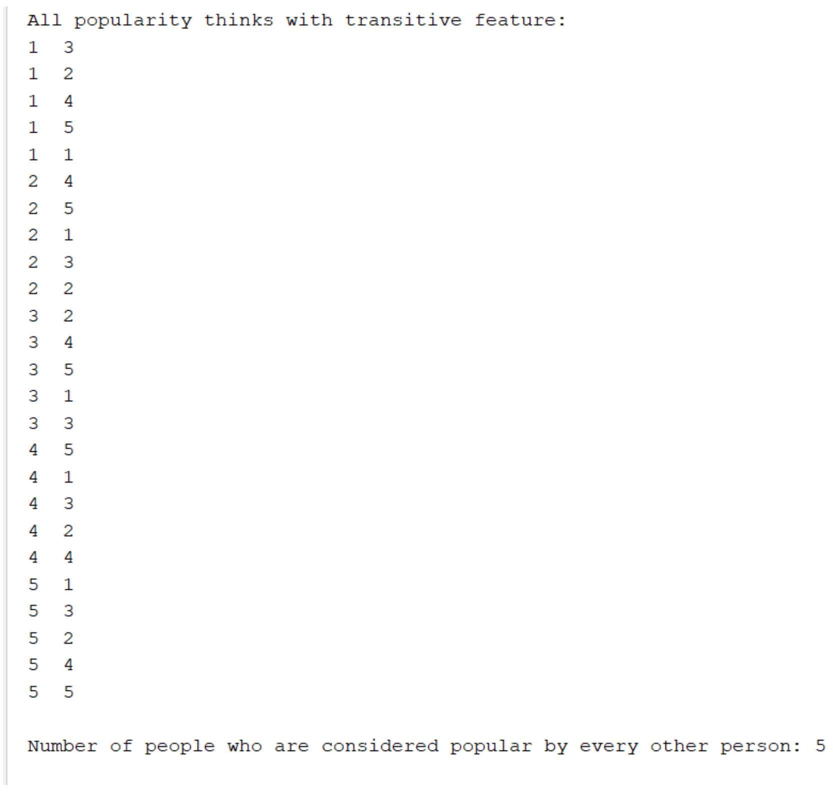
```
 Gürses\Desktop\sdföds\HOMEWORK-8\out\production\HOMEWORK-8" Main
All popularity thinks with transitive feature:
1  2
1  3
1  4
1  5
2  3
2  4
2  5
2  2
3  4
3  5
3  2
3  3
4  5
4  2
4  3
4  4
5  2
5  3
5  4
5  5

Number of people who are considered popular by every other person: 4
```

INPUT:



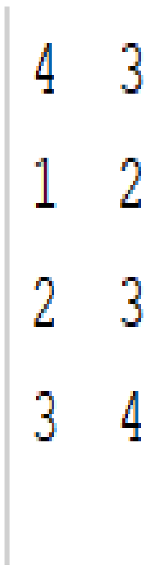
```
5 5
1 3
2 4
3 2
4 5
5 1
```

OUTPUT:

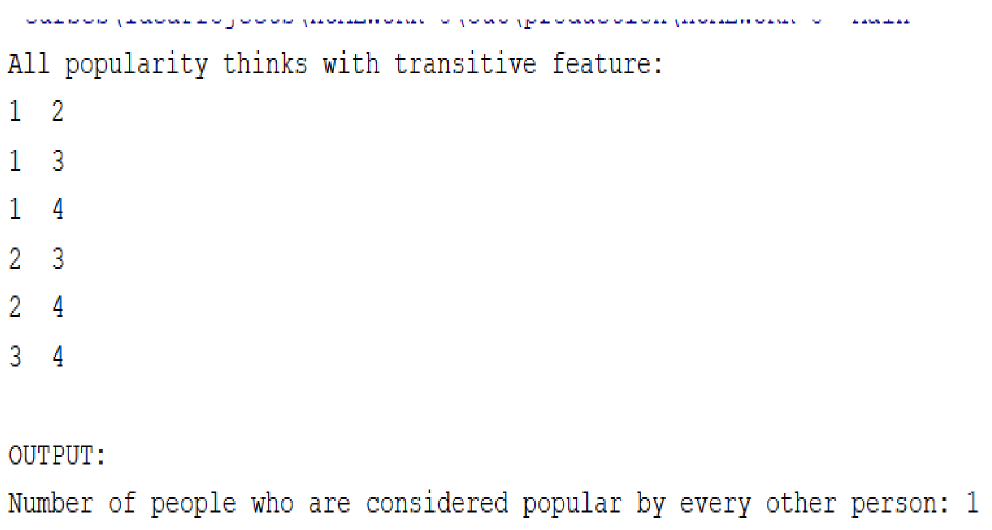


```
All popularity thinks with transitive feature:
1  3
1  2
1  4
1  5
1  1
2  4
2  5
2  1
2  3
2  2
3  2
3  4
3  5
3  1
3  3
4  5
4  1
4  3
4  2
4  4
5  1
5  3
5  2
5  4
5  5

Number of people who are considered popular by every other person: 5
```

INPUT: OUTPUT:

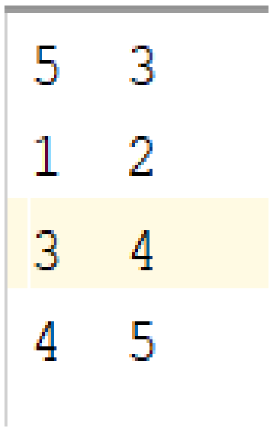


```
4   3
1   2
2   3
3   4
```

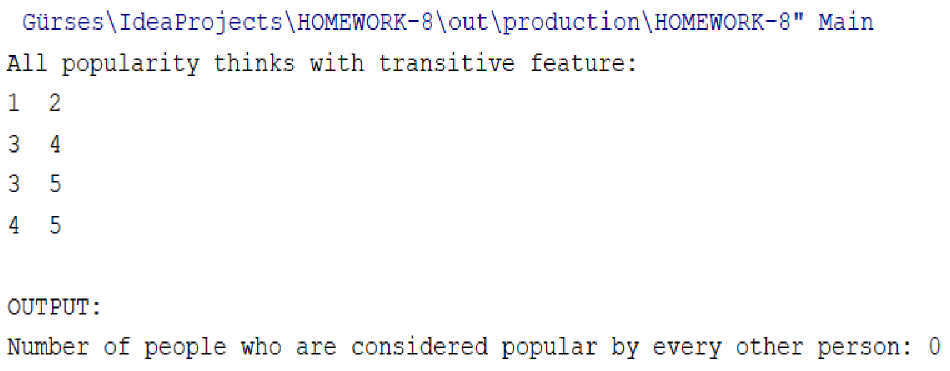


```
All popularity thinks with transitive feature:
1  2
1  3
1  4
2  3
2  4
3  4

OUTPUT:
Number of people who are considered popular by every other person: 1
```

INPUT: OUTPUT:

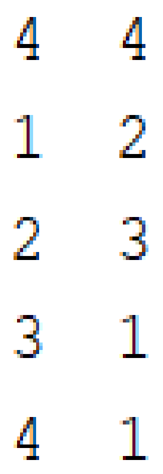


```
5   3
1   2
3   4
4   5
```

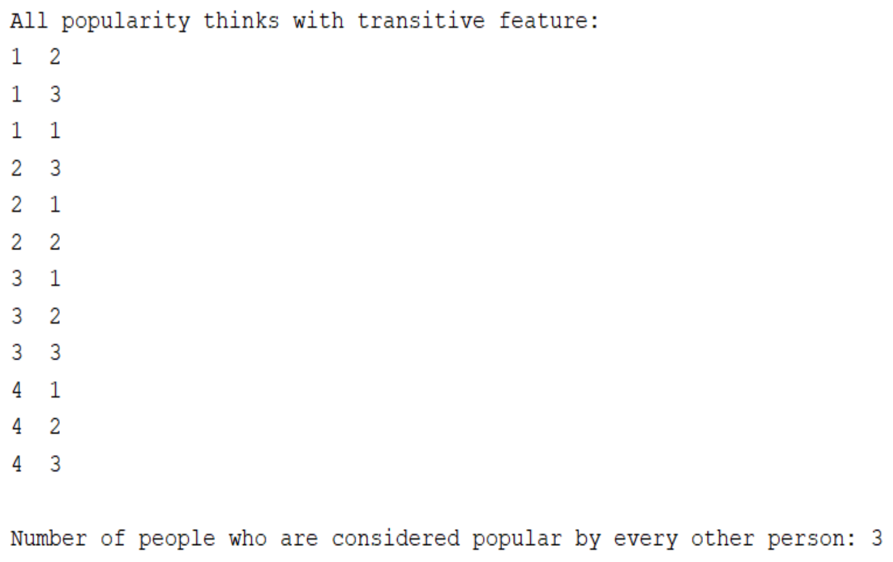


```
 Gürses\IdeaProjects\HOMEWORK-8\out\production\HOMEWORK-8" Main
All popularity thinks with transitive feature:
1  2
3  4
3  5
4  5

OUTPUT:
Number of people who are considered popular by every other person: 0
```

INPUT: OUTPUT:

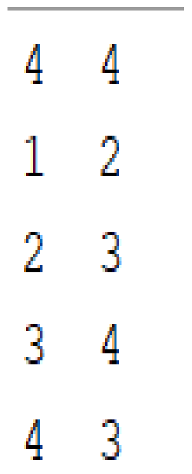


```
4 4
1 2
2 3
3 1
4 1
```

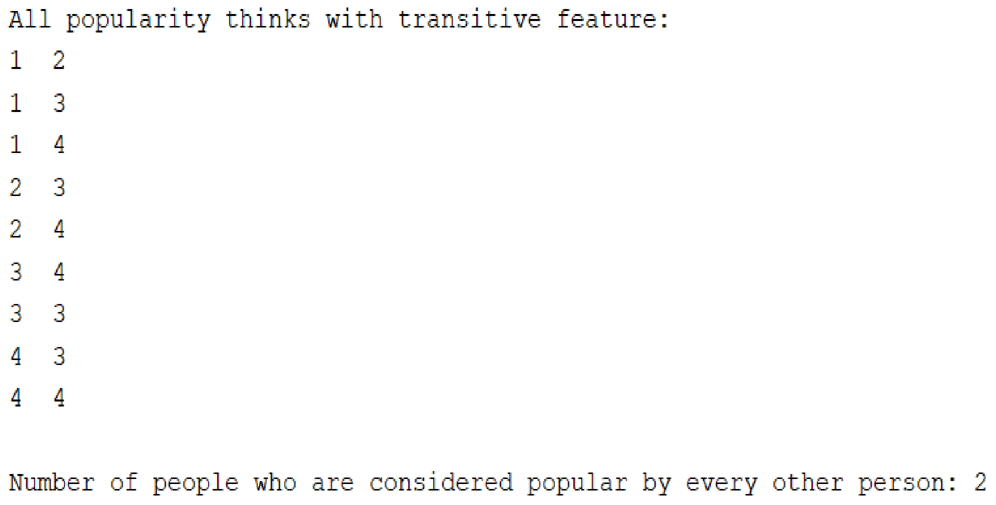


```
All popularity thinks with transitive feature:
1  2
1  3
1  1
2  3
2  1
2  2
3  1
3  2
3  3
4  1
4  2
4  3

Number of people who are considered popular by every other person: 3
```

INPUT: OUTPUT:

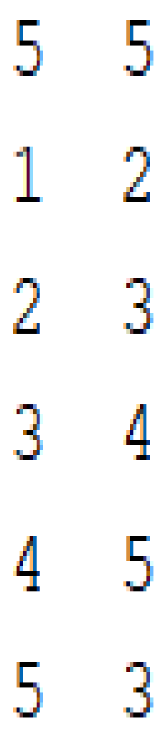


```
4 4
1 2
2 3
3 4
4 3
```

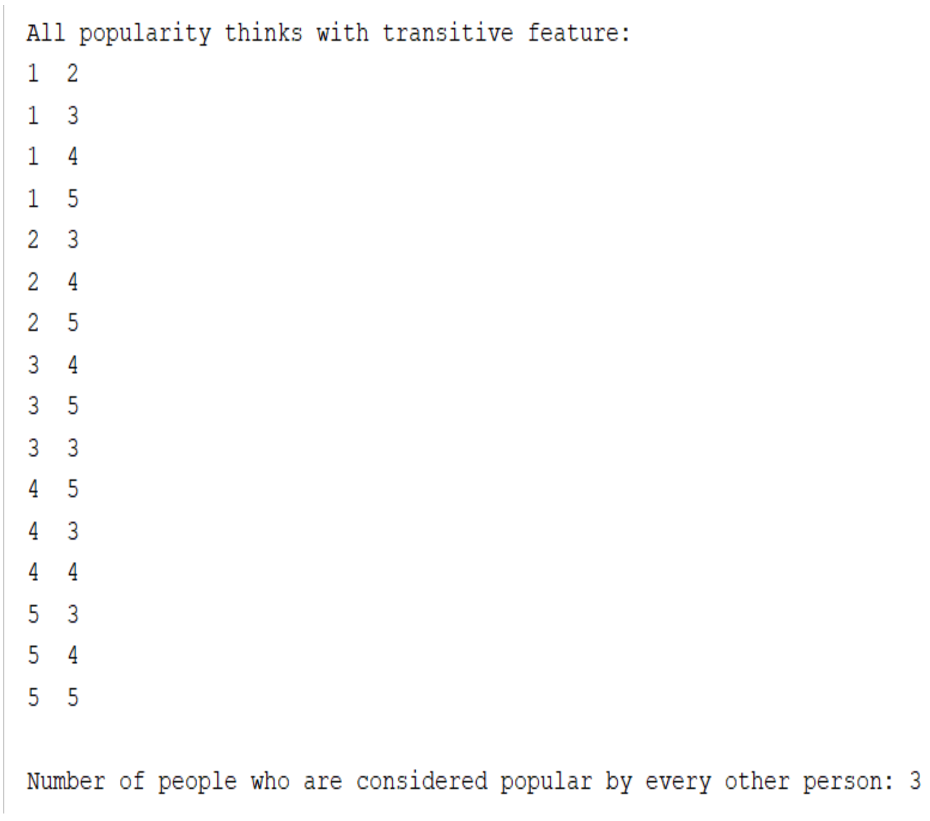


```
All popularity thinks with transitive feature:
1  2
1  3
1  4
2  3
2  4
3  4
3  3
4  3
4  4

Number of people who are considered popular by every other person: 2
```

INPUT: OUTPUT:

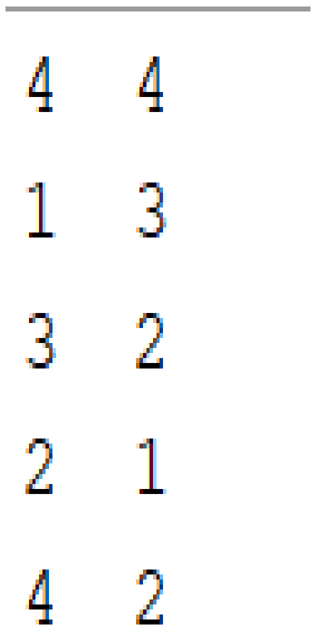


```
5 5
1 2
2 3
3 4
4 5
5 3
```

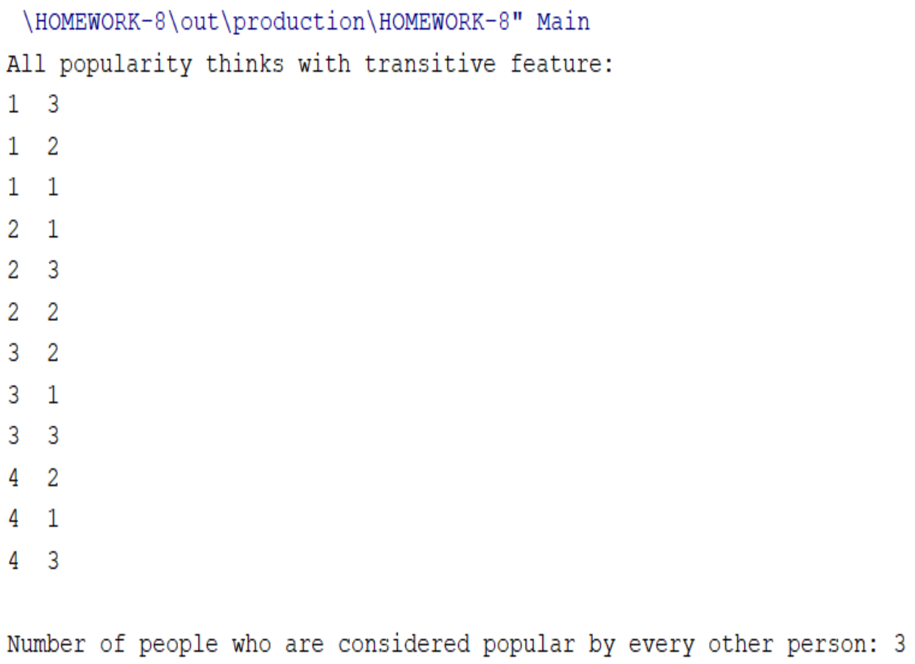


```
All popularity thinks with transitive feature:
1  2
1  3
1  4
1  5
2  3
2  4
2  5
3  4
3  5
3  3
4  5
4  3
4  4
5  3
5  4
5  5

Number of people who are considered popular by every other person: 3
```

INPUT: OUTPUT:



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

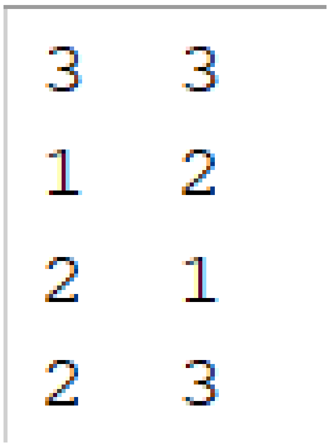


```
 \HOMEWORK-8\out\production\HOMEWORK-8" Main
All popularity thinks with transitive feature:
1  3
1  2
1  1
2  1
2  3
2  2
3  2
3  1
3  3
4  2
4  1
4  3

Number of people who are considered popular by every other person: 3
```

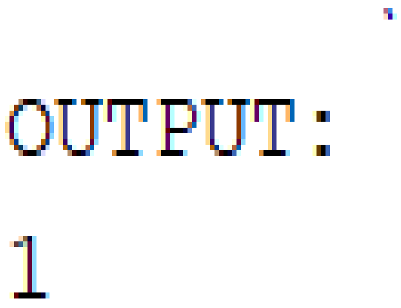
In pdf form output is only number of people who are considered popular by other person:

So output with in pdf form:

INPUT: OUTPUT:

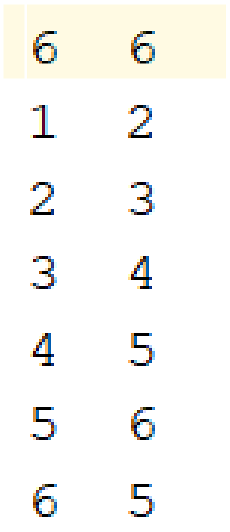


```
3    3
1    2
2    1
2    3
```

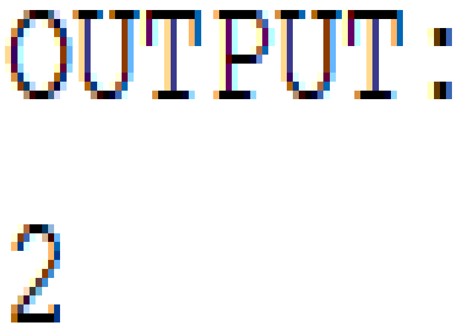


```
OUTPUT:
1
```

INPUT: OUTPUT:



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



```
OUTPUT:
2
```