

**Gebze Technical University  
Computer Engineering**

**CSE 222 - 2019 Spring**

**HOMEWORK 8 REPORT**

**Taylan ÖNDER  
151044015**

Course Assistant: Ayşe Şerbetçi TURAN

# 1 INTRODUCTION

## 1.1 Problem Definition

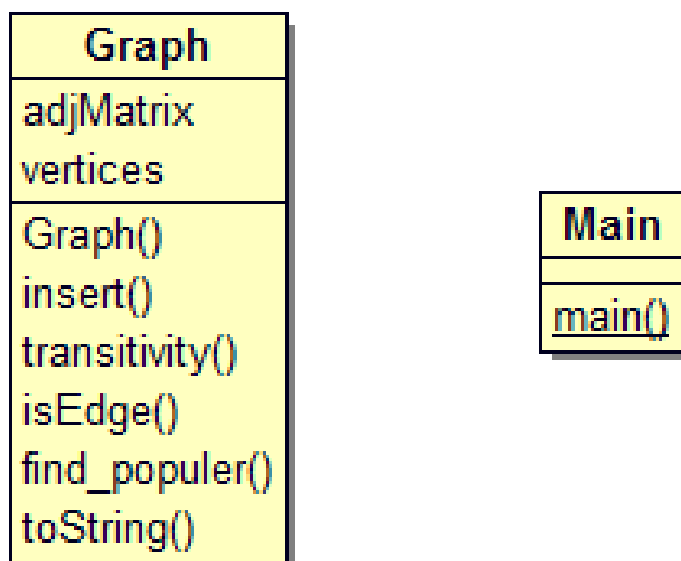
In this homework some group of people find represents the number of people who are considered popular by every other person. Some persons think that this person is popular and this relation of popularity is transitive. It means we don't have to give an input for transitive property. Describing the graph structure and calculating the total population number according to the relationship between. All pairs reading in input file. First line include vertex number and number of ordered relations. All pairs have two space-separated integers, Summary according to directed pairs program calculate an integer which represents the number of people who are considered popular by every other person.

## 1.2 System Requirements

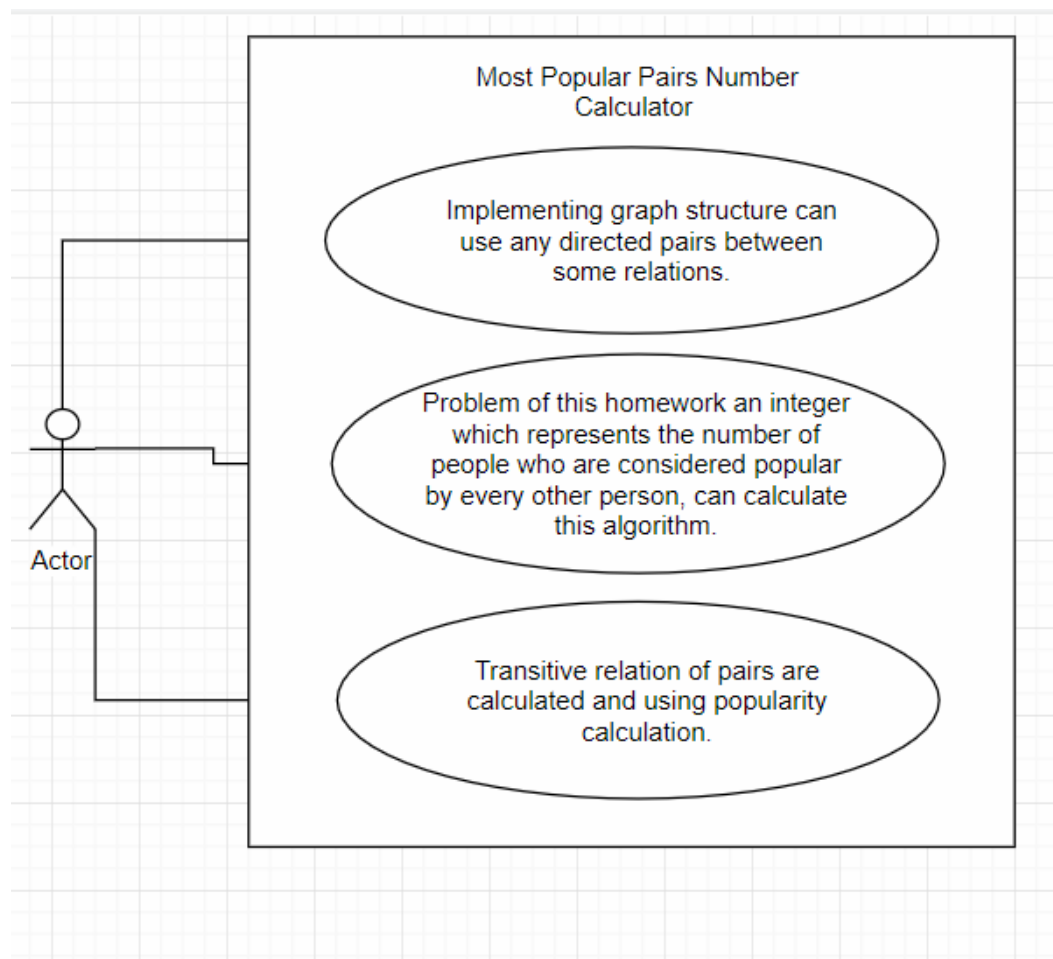
In this homework solution for problem, need to implement graph structure. Fill in the graph structure with graph insert method. Graph structure use multi dimensional boolean array. According to first line read number of people and allocate memory. For reading input file we need scanner object. For running code java code compiler required. For example intelliJ.

# 2 METHOD

## 2.1 Class Diagrams



## 2.2 Use Case Diagram



## 2.2 Problem Solution Approach

First of all I implement a graph structure to connect all pairs relation and required transitivity property. Implementing graph structure, I used Adjacency Matrix theorem. For reading input.txt, I use scanner method to read piece of pairs. I read an input file and consider N (number of people) and M (number of ordered relations). According to N number, I allocate for two dimensional boolean array to fill all pairs. After all pairs in reading file fill in my array, I called transitivity method to pairs connect together with transitivity properties. Find\_popular method find an integer which represents the number of people who are considered popular by every other person. Description of all functions which I used for homework is below.

```

public class Graph {
    private boolean adjMatrix[][];
    private int vertices;

    public Graph(int numVertices) {
        this.vertices = numVertices;
        adjMatrix = new boolean[vertices][vertices];
    }

    public void addEdge(int i, int j) {
        adjMatrix[i][j] = true;
    }

    public void transitivity() {
        int t, y, z;
        for (y = 0; y < vertices; y++) {
            for (t = 0; t < vertices; t++) {
                for (z = 0; z < vertices; z++) {
                    if (adjMatrix[t][y] == true && adjMatrix[y][z] == true)
                        adjMatrix[t][z] = true;
                }
            }
        }
    }

    public boolean isEdge(int i, int j) {
        return adjMatrix[i][j];
    }
}

```

Constructure allocate to two dimensional array for all people pairs.

According to source and destination pair add true. Comp.  $O(1)$ .

Connect method to pairs transitivity properties. Complexity  $O(n^3)$ .

Check in graph according to source and destination do we have a pair. Comp.  $O(1)$ .

```

public void find_populer() {
    transitivity();
    int i, j, counter = 0, popularity_num = 0;
    for (i = 0; i < vertices; i++) {
        counter = 0;
        for (j = 0; j < vertices; j++) {
            if (isEdge(j, i) && j != i) {
                counter++;
            }
        }
        if (counter >= vertices - 1) {
            popularity_num++;
        }
    }
    System.out.println(popularity_num);
}

public String toString() {
    StringBuilder s = new StringBuilder();
    for (int i = 0; i < vertices; i++) {
        s.append(i + 1 + ": ");
        for (boolean j : adjMatrix[i]) {
            s.append((j ? 1 : 0) + " ");
        }
        s.append("\n");
    }
    return s.toString();
}

```

Find method is calculator of max popular people number Complexity  $O(n^3)$ .

Optional written method to show user adding pairs and transitivity status.

## 3 RESULT

### 3.1 Test Cases

Firstly reading file and find Two space-separated integers, N (number of people) and M (number of ordered relations).After that allocate memory with graph constructure to create graph structure.Continue to read to fill pairs in this structure.After all pairs filling, again fill structure according to pairs transitivity properties.After the last filling calculate An integer which represents the number of people who are considered popular by every other person with find\_popular method and print result to screen. Below are a few scenarios I tried ;

**Note :** I consider destiation and source are the same situation when adding and finding pairs transivities.

Input



Dosy

3 3

1 2

2 1


2 3

Output

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

1 is an integer which represents the number of people who are considered popular by every other person.

Input

 inp

Dosya


```
6 7
1 2
2 3
4 1
4 3
6 4
5 6
3 4
```

Output

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

4 is an integer which represents the number of people who are considered popular by every other person.

Input

 inp

Dosya

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

Output

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

2 is an integer which represents the number of people who are considered popular by every other person.