Gebze Technical University
Computer Engineering

CSE 222 - 2018 Spring

HOMEWORK 8 REPORT

GÖKÇE NUR ERER
171044079

Course Assistant: Ayşe Şerbetçi Turan

1 INTRODUCTION

1.1 Problem Definition

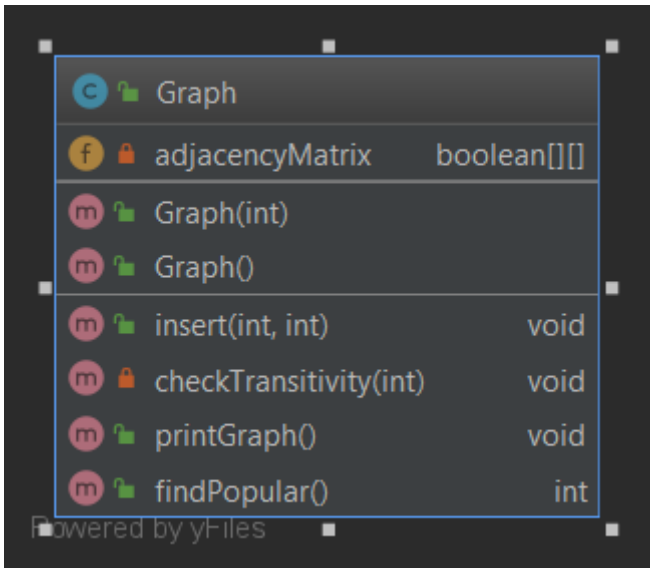
In this assignment, we have a group of people in which have an ordered popularity relation between person pairs. If P1 and P2 has a relation that means P1 thinks P2 is popular. This relation is also transitive which means if the relations P1,P2 and P2,P3 exists the relation P1,P3 should exist too. The goal of this assignment is to calculate the number of people who are considered popular by every other person.

1.2 System Requirements

This assignment can run on any average computer which has Java JVM and has an IntelliJ Idea IDE since this assignment is delivered as an IntelliJ project.

2 METHOD

2.1 Class Diagrams



2.2 Use Case Diagrams

This software reads an input from a .txt file called input.txt so there is nothing expected from user. If the user wants to see the adjacency matrix of the graph, by entering 1 to the program when asked, they can view it. Entering something else causes the program to terminate since there is nothing else to run.

2.3 Problem Solution Approach

To solve this problem the people and their relations can be represented with a graph. So in this project a class called Graph implements a graph with an adjacency matrix which has a type of boolean. This class and its methods are explained below:

public Graph(int vertexNum) :	constructs a graph instance with an adjacency matrix of size vertexNum * vertexNum
public Graph():	constructs an empty graph
public void insert(int source , int destination):	Inserts an edge to the graph by setting the related cell (adjmat[source][destination]) in the adjacency matrix true. After the insert the transitivity relations between the edges are checked using the checkTransitivity method. The time complexity of this method is $O(n^3)$ since the loop runs for n times, n being the number of the vertices and since the method checkTransitivity has a time complexity of $O(n^2)$ they get multiplied so the complexity of this method is $O(n*n^2) = O(n^3)$.
private void checkTransitivity(int source):	Checks if a certain source causes a transitive relation and sets according cells which has to be setted true. The time complexity of this method is $O(n^2)$, n being the number of vertices since there are 2 nested loops and at the worst case each loop run n times the time complexity is $O(n*n) = O(n^2)$.
public void printGraph():	Prints the adjacency matrix of the of the graph. The time complexity of this method is

	$O(n+n^2) = O(n^2)$ since there are 2 nexted loops which runs n times, n being the number of vertices. Also the first loop which prints the person numbers runs n times so it causes $n + n^2$ in the previous expression.
public int findPopular():	Finds the number of people who are considered popular by every other person by keeping every person's total popularity consideration count in an array. By comparing each array element with number of vertices -1 the result is found. The time complexity of this method is $O(n+n^2) = O(n^2)$ since there are 2 nexted loops which runs n times, n being the number of vertices. Also the first loop which prints the person numbers runs n times so it causes $n + n^2$ in the previous expression.

To construct a graph to solve this problem, the input is taken from a txt file. So all the file readings is done in the Main class, in the main method.

PS: In this assignment it is considered that the people themselves do not think themselves as popular. For example if 2 thinks 1 is popular and 1 think 2 is popular, 2 doesn't think that itself is popular.

3 RESULT

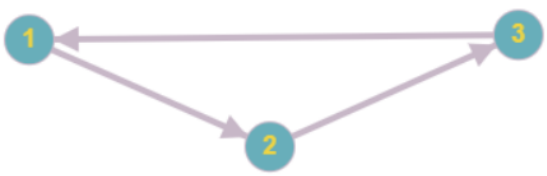
3.1 Test Cases

When showing the graphs, the transitivity relations of the people are not shown to show the given input data. Transitive results are in the adjacency matrix.

Case 1:

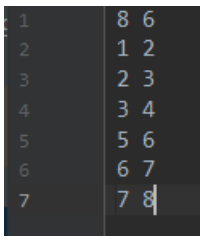
Case 2:

Case 3:



3.2 Running Results

Case 1:




```
The number of people who are considered popular by every other person:0
Enter 1 to check the adjacency matrix:
0

Process finfinished with exit code 0
```

Text file

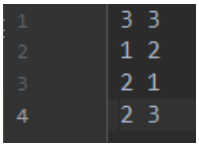
Output without the adjacency matrix

```
The number of people who are considered popular by every other person:0
Enter 1 to check the adjacency matrix:
1
      1       2       3       4       5       6       7       8
1   false   true    true    true    false   false   false   false
2   false   false   true    true    false   false   false   false
3   false   false   false   true    false   false   false   false
4   false   false   false   false   false   false   false   false
5   false   false   false   false   false   true    true    true
6   false   false   false   false   false   false   true    true
7   false   false   false   false   false   false   false   true
8   false   false   false   false   false   false   false   false

Process finished with exit code 0
```

Output with the adjacency matrix

Case 2:




```
The number of people who are considered popular by every other person:1
Enter 1 to check the adjacency matrix:
0

Process finished with exit code 0
```

Text file

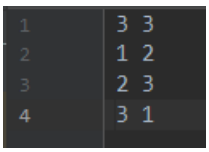
Output without the adjacency matrix

```
The number of people who are considered popular by every other person:1
Enter 1 to check the adjacency matrix:
1
      1       2       3
1   false   true    true
2   true    false   true
3   false   false   false

Process finished with exit code 0
```

Output with the adjacency matrix

Case 3:



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

Text file

```
The number of people who are considered popular by every other person:3
Enter 1 to check the adjacency matrix:
0

Process finished with exit code 0
```

Output without the adjacency matrix

```
The number of people who are considered popular by every other person:3
Enter 1 to check the adjacency matrix:
1
      1       2       3
1   false   true    true
2   true    false   true
3   true    true    false

Process finished with exit code 0
```

Output with the adjacency matrix

- Main titles -> 16pt , 2 line break
- Subtitles -> 14pt, 1.5 line break
- Paragraph -> 12pt, 1.5 line break