

**Gebze Technical University
Computer Engineering**

CSE 222 - 2019 Spring

HOMEWORK 8 REPORT

**AHMET MELİH YANALAK
151044044**

Course Assistant:AYŞE ŞERBETÇİ TURAN

1 INTRODUCTION

1.1 Problem Definition

In this project programmers are asked to read all the relations between two people that given through input file and keep them in a proper data structure. In addition, some of the relations that is not specified in the input file must be included by the transitivity rule. In more details, if there is a connection between A and S ($A \rightarrow S$) which means that "A thinks S is popular" and there is a connection between S and F ($S \rightarrow F$) then even if a relation not specified such that ($A \rightarrow F$), it must take part in relations.

The problem is to find the number of people popular by everyone.

1.2 System Requirements

This project requires a hardware with $|V|^2$ bits memory, operation system that can run IntelliJ Idea Community Edition 2018 3.5. and libraries;

Java.util.Iterator

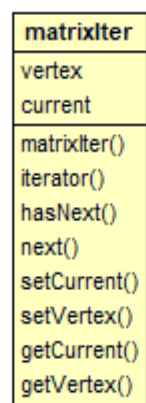
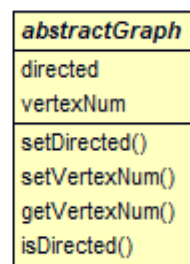
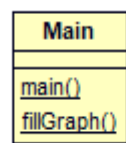
Java.util.BufferedReader

Java.util.FileReader

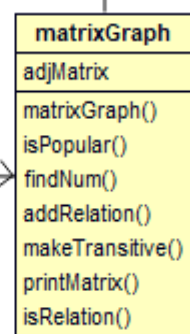
Java.util.IOException

2 METHOD

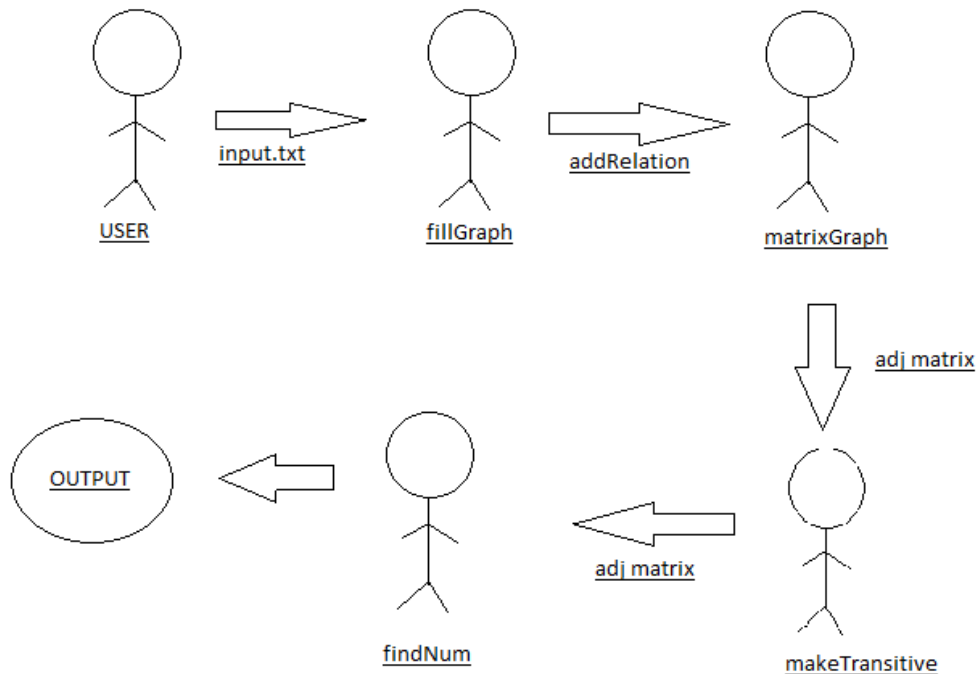
2.1 Class Diagrams



mg



2.2 Use Case Diagrams

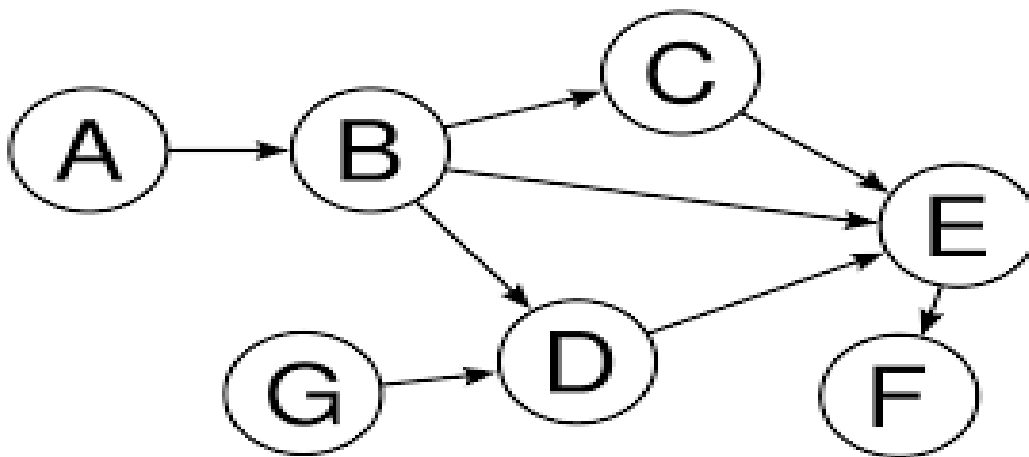


2.3 Problem Solution Approach

First of all, before beginning to solve the problem, a proper data structure should have been chosen by the programmer which will lead him/her to the solution.

In this project, since we are dealing with connections of pairs of people, Graph Data Structure might be a good choice. In addition, the connections of pairs are always single directional. Which means that using directional graph will be quite useful for that solution.

The directional graph will be used in project could be implemented by using adjacency matrix or adjacency list. I worked with adjacency matrix which is easier for me to implement by using java array.



First of all, the program should read the input file line by line to add connections to empty graph. The first line is always consist of two integer value (Number of people and number of connections between them). After reading first line, the program creates an empty graph then start reading relations one by one and add them to the graph.

After the insertion is done, before the calculating the number of popular people, transitive relations must be added to the graph which is handled by makeTransitive method. In details, it traverse all the people and if there is a relation between A and S, it checks all the people which is connected to A and make them connect to S. Also checks all the people S is connected, then make A is also connected to them.

Finally, in order to determine a person if popular or not, adjacency matrix should be checked. If a column is full of connection except the person itself, which means that person is popular. It uses this algorithm for each person and finds the number of people popular by everyone.

Due to using adjacency matrix, there will be $|V|$ rows and $|V|$ columns if $|V|$ is considered as the number of people. which means that there will be $|V|^2$ boolean values. The space complexity of this program is $O(|V|^2)$.

THE TIME COMPLEXITIES OF THE METHODS

isPopular -> $O(|V|)$ because of the checking the column if it's full of true or not.

findNum -> $O(|V|^2)$ For all people, calls the isPopular method

addRelation -> $O(1)$ Only changes the value in the specified index.

makeTransitive -> $O(|V|^3)$ for each boolean value($O(|V|^2)$) controls the connections in a loop.

3 RESULT

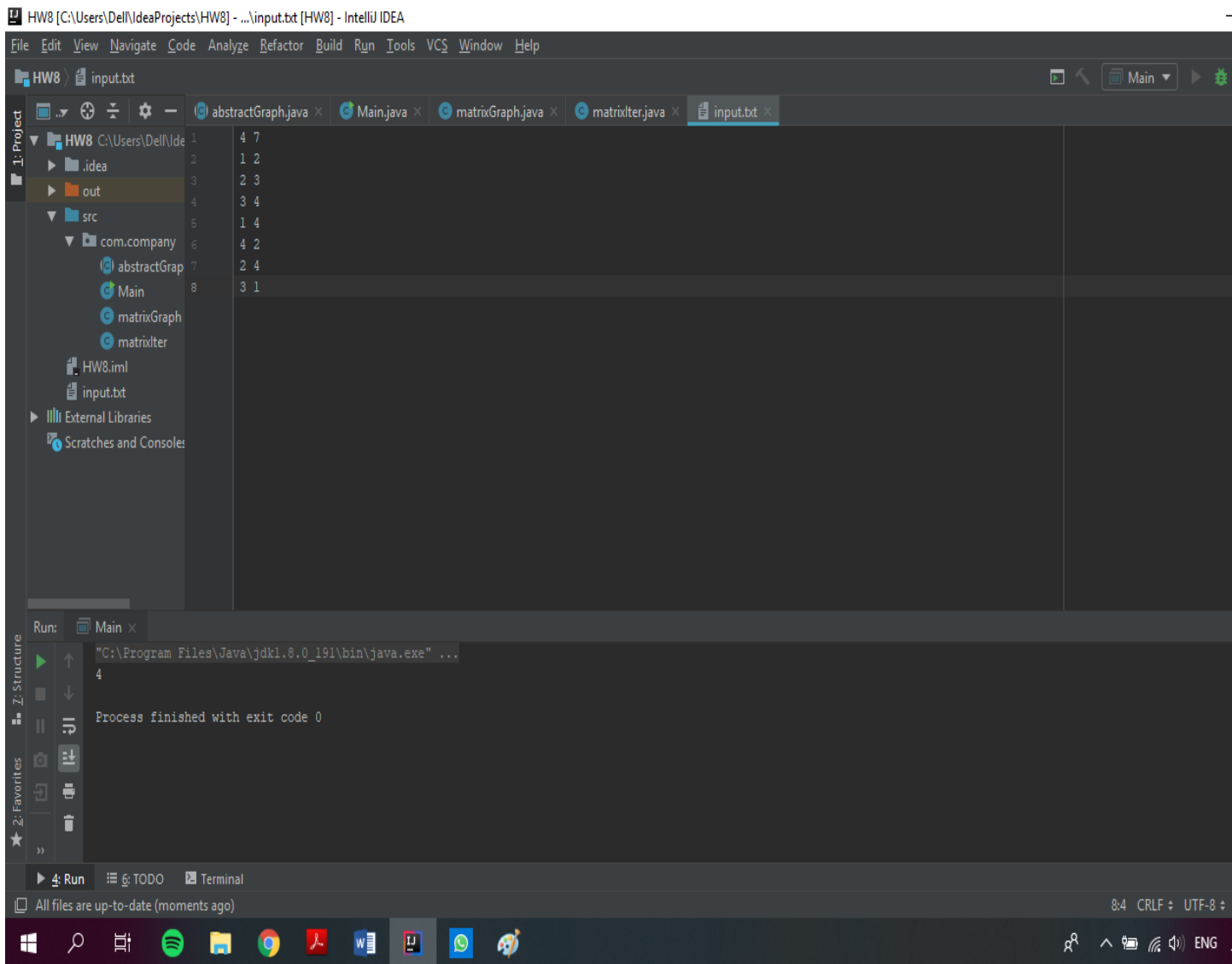
3.1 Test Cases

The program is tested by using different input files that consist of the situations;

- 1) Each person is popular
- 2) Nobody is popular
- 3) There is only one person

TEST RESULTS IS BELOW:

3.2 Running Results



The screenshot displays the IntelliJ IDEA interface for a project named "HW8". The main editor window shows the content of the file "input.txt", which contains a 4x7 grid of numbers:

```
4 7
1 2
2 3
3 4
1 4
4 2
2 4
3 1
```

The Run window at the bottom shows the execution of the "Main" class. The command executed is:

```
"C:\Program Files\Java\jdk1.8.0_191\bin\java.exe" ...
```

The output of the run is:

```
4
Process finished with exit code 0
```

The status bar at the bottom indicates that all files are up-to-date (moments ago) and the encoding is UTF-8.

The screenshot shows an IDE interface with the following components:

- File Explorer (Left):** Displays the project structure for 'HW8'. The 'src' directory is expanded, showing the package 'com.company' with classes 'abstractGraph', 'Main', 'matrixGraph', and 'matrixIter'. Other files include 'HW8.iml' and 'input.txt'.
- Code Editor (Center):** Shows the content of 'input.txt' with the following text:

```
6 8
1 2
2 3
4 1
4 3
6 4
5 6
5 1
3 4
```
- Run Console (Bottom):** Shows the execution of the 'Main' class. The command executed is `"C:\Program Files\Java\jdk1.8.0_191\bin\java.exe" ...`. The output is `4`, and the message 'Process finished with exit code 0' is displayed.

HW8 [C:\Users\Dell\IdeaProjects\HW8] - ...input.txt [HW8] - IntelliJ IDEA

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

HW8 > input.txt

1: Project

HW8 C:\Users\Dell\IdeaProjects\HW8
├── .idea
├── out
└── src
 ├── com.company
 │ ├── abstractGraph.java
 │ ├── Main.java
 │ ├── matrixGraph.java
 │ └── matrixIter.java
 ├── HW8.iml
 └── input.txt
├── External Libraries
└── Scratches and Consoles

abstractGraph.java x Main.java x matrixGraph.java x matrixIter.java x
1 3 1
2 1 2

Run: Main x

2: Structure

"C:\Program Files\Java\jdk1.8.0_191\bin\java.exe" ...
0
Process finished with exit code 0

4: Run

6: TODO Terminal

All files are up-to-date (moments ago)



HW8 [C:\Users\Dell\IdeaProjects\HW8] - ...input.txt [HW8] - IntelliJ IDEA

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help



1: Project

The screenshot shows the Project Explorer in IntelliJ IDEA. The project is named 'HW8' and is located at 'C:\Users\Del\Idea'. The structure is as follows:

- HW8
 - .idea
 - out
 - src
 - com.company
 - abstractGraph
 - Main
 - matrixGraph
 - matrixIter
 - HW8.iml
 - input.txt
- External Libraries
- Scratches and Consoles

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

Run: Main x

★ 2: Favorites ■■ 7: Structure

► 4: Run ≡ 6: TODO >_ Terminal

☐ All files are up-to-date (moments ago)



HW8 [C:\Users\Dell\IdeaProjects\HW8] - ...input.txt [HW8] - IntelliJ IDEA

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

HW8 > input.txt

1: Project

HW8 C:\Users\Dell\IdeaProjects\HW8

.idea

out

src

com.company

abstractGraph.java

Main.java

matrixGraph.java

matrixIter.java

HW8.iml

input.txt

External Libraries

Scratches and Consoles

abstractGraph.java

Main.java

matrixGraph.java

matrixIter.java

```
1 6 7
2 1 2
3 2 3
4 4 1
5 4 3
6 6 4
7 5 6
8 3 4
```

Run: Main x

"C:\Program Files\Java\jdk1.8.0_191\bin\java.exe" ...

4

Process finished with exit code 0

2: Favorites

Run

Up

Down

Pause

Refresh

Search

Print

Close

More

4: Run 6: TODO Terminal

All files are up-to-date (moments ago)

