Gebze Technical University
Computer Engineering

CSE 222 - 2018 Spring

HOMEWORK 8 REPORT

MELİH YABAŞ
161044072

Course Assistant: AYŞE ŞERBETÇİ TURAN

1 INTRODUCTION

1.1 Problem Definition

There are some people which an ordered popularity relation. (|A->B| A thinks B is popular) We are searching some transitive relations. The relation is transitive which means that if the relations (P1,P2) and (P2,P3) exist, than (P1,P3) also exist event if it is not specified by the input pairs.
We must find the people who are popular by every other person.

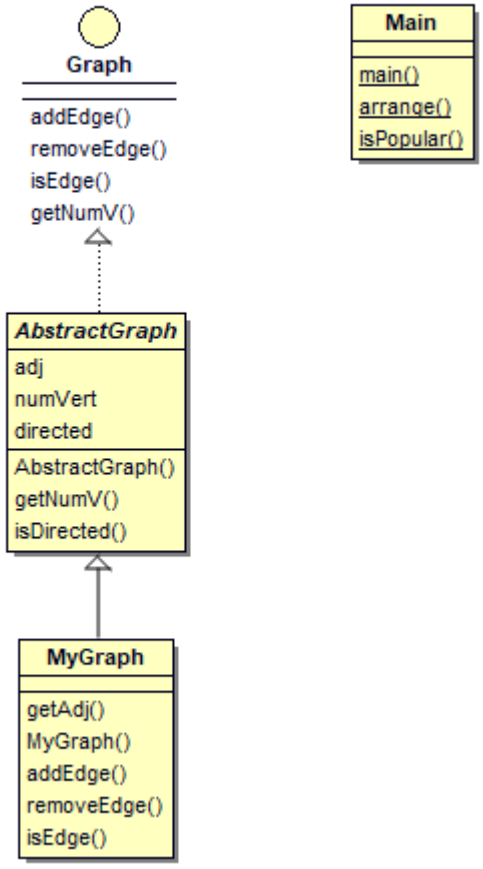
1.2 System Requirements

This program needs Java Virtual Machine to work properly.
Java Virtual Machine Requirements:
Windows 10/8/7/Vista/XP/2000
Windows Server 2008/2003 4
Intel and 100% compatible processors are supported.
Pentium 166 MHz or faster processor wit at least 64 MB of physical RAM. 6
98 MB for free disk space.

To run the program,
We need
1,2 MB free disk space
128 KB ram

2 METHOD

2.1 Class Diagrams



2.2 Problem Solution Approach

Firstly, we need to create a graph class. To apply object oriented principle correctly, we must create an interface and an Abstract class.

I took the first number of the file to create the graph. I added the nodes into to the graph from the file. I keep the relation in a 2 dimentional array. (adjacency matrix)

Then I wrote a method to traverse the graph.(arrange method) This method takes the adjacency matrix and traverse it to make its index 0 to 1, if it is necessary.

The other method (isPopular) takes a person and control if it is popular by all people.

Using a for loop, I control all the nodes and print the people number that popular by all people.

Time Complexity of Methods:

addEdge():
O(1) because I just assign true to the given index.
removeEdge():
O(1) because I just assign false to the given index.
isEdge():
O(1) because I just return the given index.
arrange():
V the is number of vertices. O(V^3)
isPopular():
V the is number of vertices. O(V)

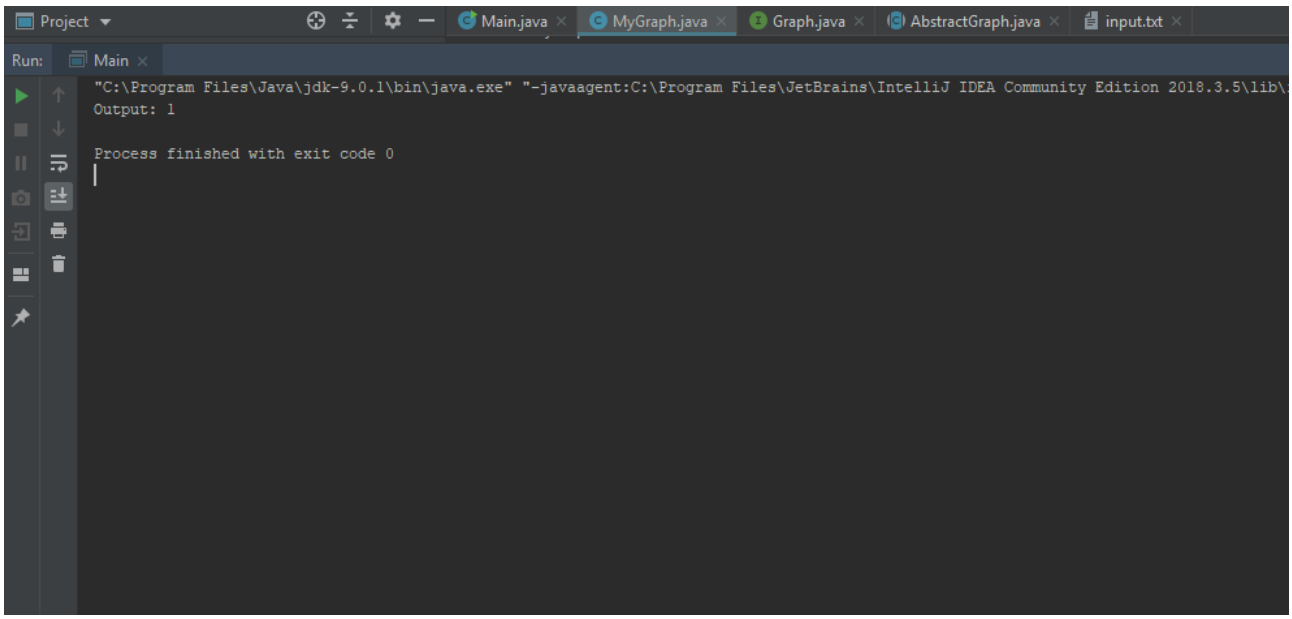
Space Complexity:
Space Complexity is O(1) for all the methods.

3 RESULT

3.1 Test Cases

First, I test my program for sample input file. Then I test it for different numbers and vertices. Some of these data have a few nodes and many relations. And some of them have many nodes and a lot of relations etc. These different data show us the program actually works.
To decide the program actually works, we must try all the exceptive situations. Then we can say that it works.

3.2 Running Results


Project
Main.java
MyGraph.java
Graph.java
AbstractGraph.java
input.txt
Run:
Main
"C:\Program Files\Java\jdk-9.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2018.3.5\lib\
Output: 1
Process finished with exit code 0