HACETTEPE UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

BBM203 PROGRAMMING LAB.
ASSIGNMENT 1

Subject : Data Structures (Graph) and GUI
Submission Date : 19.10.2012
Deadline : 05.11.2012, 10:00 am
Programming Language : JAVA (JDK7)
Advisors : Dr. Sevil ŞEN, Dr. Burcu CAN, R. A. Ali Osman SERHATOĞLU

1. INTRODUCTION / AIM:

In this experiment, you are expected to develop a simple graphic program, UndirectedGraph. The main aim of the experiment is to make you get familiar with data structures and practice graphical user interfaces (GUI) with Java programming language and Java's GUI components included within javax.Swing package.

Other aims of the experiment are to make you use computers to solve mathematical problem, to use exception handling in case of errors occurred during the execution of the program and also to perform file operations by using input/output (io) libraries of Java.

2. BACKGROUND INFORMATION

2.1 GUI

A graphical user interface (GUI) is one of the most important parts of a program. It is the layer providing interaction with the user, by listening the user's actions which mean commands, and displaying the changes after those commands are performed. There are several ways of designing GUI with Java programming language such as applets, swing components, etc. The main aim of this programming assignment is to make you practice GUI components of Java environment imported with javax.Swing package.

2.2 Graph

In mathematics, a graph is an abstract representation of a set of objects where some pairs of the objects are connected by links. The interconnected objects are represented by mathematical abstractions called vertices, and the links that connect some pairs of vertices are called edges.

- A graph G is a pair (V,E), where V is a finite set and E is a binary relation.
- The set V is called the vertex set of G, and its elements are called vertices.
- The set E is called the edge set of G, and its elements are called edges.

There are two kinds of graphs: directed and undirected.

Undirected Graph

If (u,v) is an edge in an undirected graph G=(V,E), we say (u,v) is incident on vertices u and v.

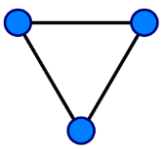


Figure 1: Undirected Graph

Directed Graph

If (u,v) is an edge in a directed graph G=(V,E), we say that (u,v) is incident from or leaves vertex u and is incident to or enters vertex v.

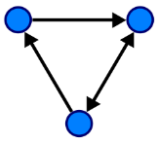


Figure 2: Directed Graph

In this experiment you will use undirected graph.

2.3 File Operations in JAVA

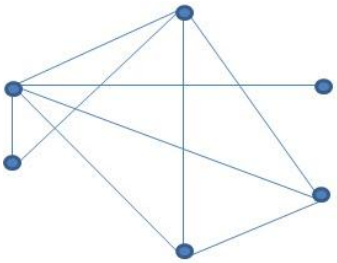
Java provides a rich API for file operations. It has a large library for different file operations. In this experiment you will use IO (Input/Output) classes of Java to perform file operations.

3. EXPERIMENT

3.1 Problem

You will have a list of edge numbers which belongs to vertexes in this assignment. Every vertex (vertices) is a point which is nonlinear with other points. You supposed to draw a graph with this information if there is a graph with these degrees.

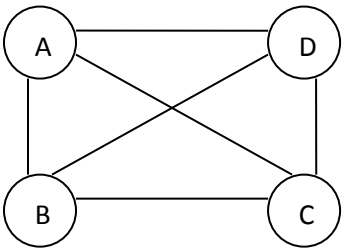
For Example ➔ 5 4 3 3 2 1



You should design an algorithm for this problem.

Other problem of the experiment is to find number of triangles which are in the graph.

For Example ➔ 3 3 3 3 => ACD, ABD, ABC, DBC => 4



You should design an algorithm for the other problem too.

3.2 Program

UndirectedGraph, the software you are supposed to develop for this experiment.

Command-Line Arguments:

UndirectedGraph.jar <outputfilename> [<integers>]

<integers> ➔ a list of edge numbers

Input:

The program should accept input with GUI or command-line. Command-line argument example is shown precious section. GUI example is shown below:

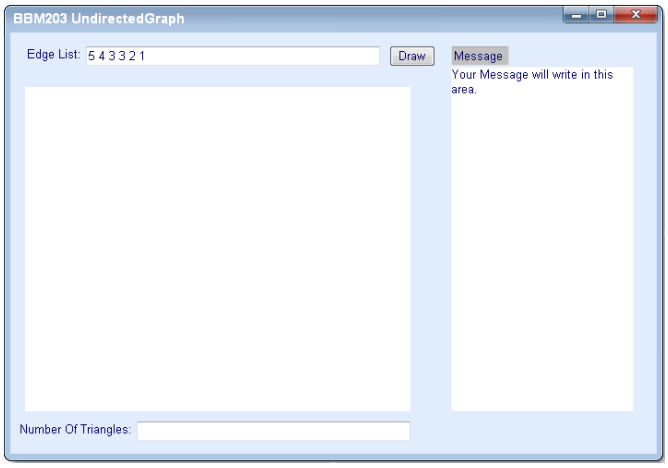
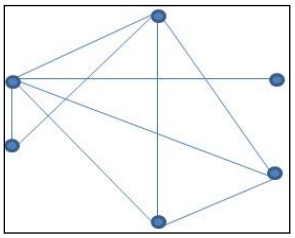


Figure 3: UndirectedGraph GUI

Output:

outputfilename<numberofoutput>.jpg ➔ Shape of Graph.

For example => output1.jpg



outputfilename<numberofoutput>.txt ➔ Edge of Graph().

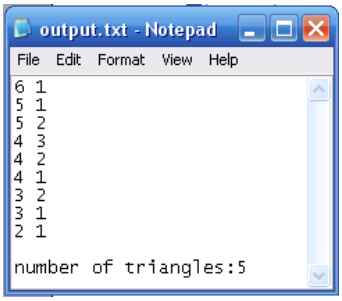
Vertex id is given like below:

Edge List From Input : 5 4 3 3 2 1
Vertex Id : 1 2 3 4 5 6
Format:

```
edge => <biggestvertexid>       <biggestothervertexid>
edge => <biggervertexid>        <biggerothervertexid>
.
.
.
edge => <smallestvertexid>      <smallestothervertexid>

number of triangles:<numberoftriangle>
```

For example => output1.txt



Error Format:

All caught exceptions will be written to an output file(txt) with their error codes. Here is the list of the exceptions which have to be caught and written to the errors in the file:

➔ 0001 : The graph is cannot drawn
➔ 0002 : Invalid input

All of the other exceptions which may be caught will be written as:

➔ 0003 : Unknown error!

3.3 Execution Of The Program

User launches the program, enters edge numbers of vertexes and clicks the “Draw” button. An example is shown below:

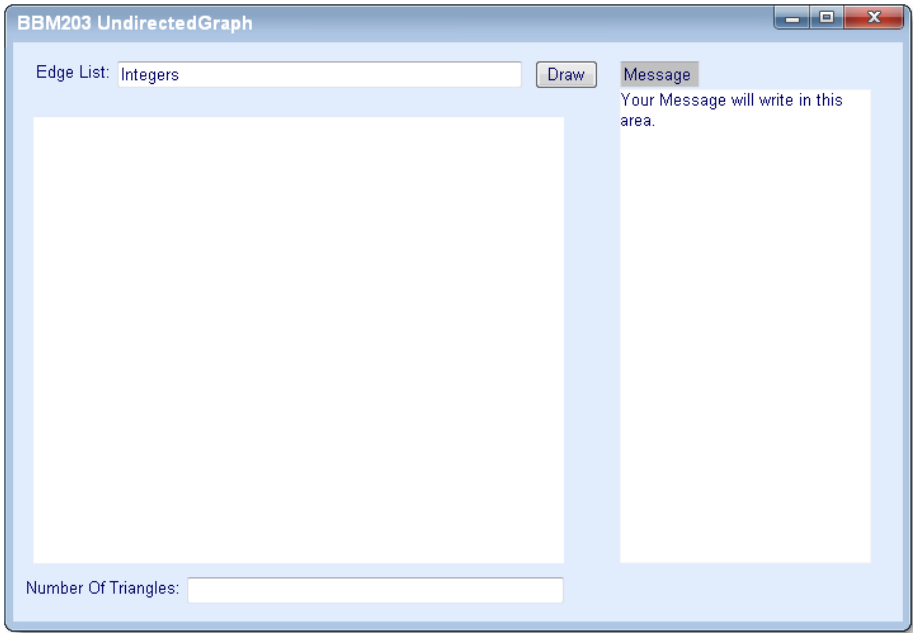


Figure 4: Start of Program

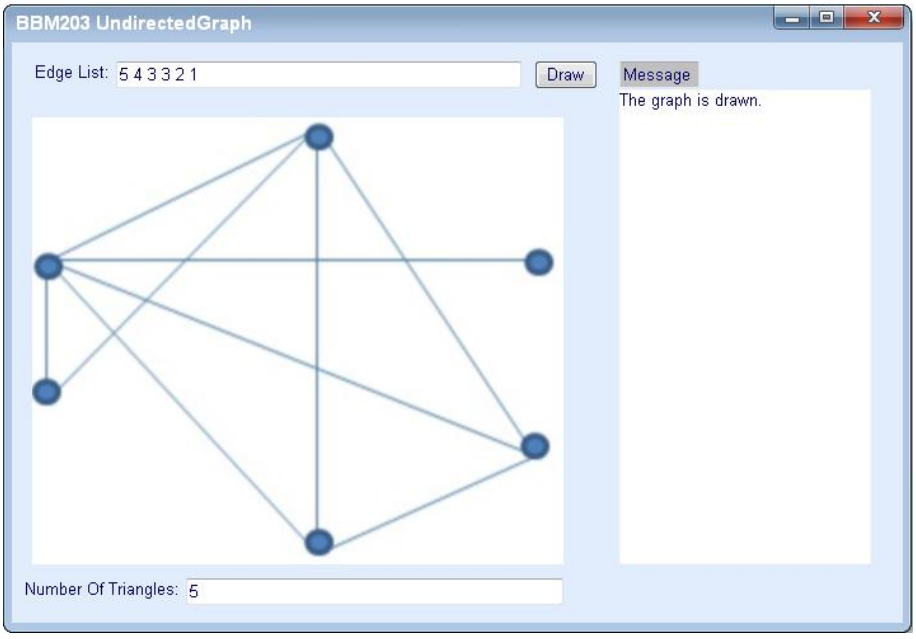


Figure 5: After The Click

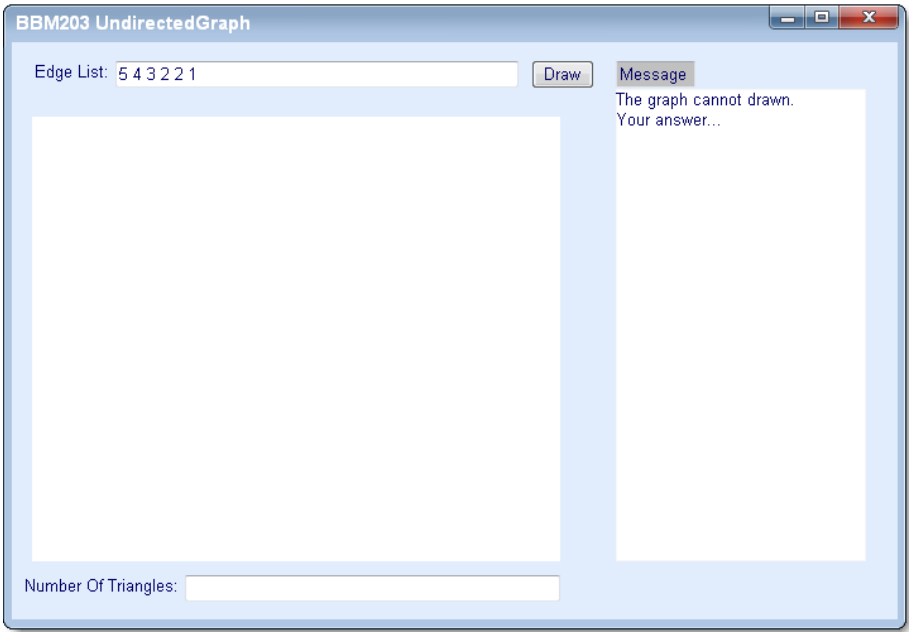


Figure 6:Error in GUI

3.4 Your Report

You also should evaluate time-space complexity of the two problem in your report.

LAST REMARKS:

- Use Eclipse IDE for JAVA Developers as development environment. Any other development enviroments will not be accepted!
- Regardless of the length, use **UNDERSTANDABLE** names to your variables, classes and functions.
- Write **READABLE SOURCE CODE** block.
- Your submission code file structure is going to be announced later.
- **You will use online submission system to submit your experiments. https://submit.cs.hacettepe.edu.tr/ Deadline is: 10:00 am. No other submission method (such as diskette, CD or email) will be accepted.**
- Do not submit any file via e-mail related with this assignment.
- **SAVE** all your work until the assignment is graded.
- The assignment must be original, **INDIVIDUAL** work. Duplicate or very similar assignments are both going to be punished. General discussion of the problem is allowed, but **DO NOT SHARE** answers, algorithms or source codes.
- You can ask your questions through course's news group and you are supposed to be aware of everything discussed in the newsgroup: **news://news.cs.hacettepe.edu.tr/dersler.bbm203**

REFERENCES

- T. Cormen, C. Leiserson, R. Rivest, C. Stein, Introduction to Algorithms, Second Edition, The MIT Press, 2003 (Clrs)
- H. M. Deitel, P. J. Deitel, Java How to Program, Prentice Hall
- http://examples.oreilly.com/jswing2/code/
- http://docs.oracle.com/javase/tutorial/uiswing/examples/components/index.html
- http://www.eclipse.org/downloads/