

Project #1 (due April 24, 10pm )

Lecturers: Sanem Sariel, Serif Bahtiyar

Code - 60 pts

You are given two input file : **input1.txt**, **input2.txt**.

**input1.txt** contains adjacency matrix of the **undirected** graph as illustrated in Figure 1.  $i^{th}$  line of the file give information about the connectivity of  $i^{th}$  node in the graph. This input file will be used for the questions (a) and (b).

**input2.txt** contains adjacency matrix of the **directed** graph.  $i^{th}$  line of the file give information about the connectivity of  $i^{th}$  node in the graph. Please pay attention that this matrix is **not symmetric**. This input file will be used for the question (c).

- (a) (15 pts) Implement an efficient algorithm to compute **the number of shortest path(s) from  $s$  to  $t$**  in  $G$ . The length of a path is the number of edges on it, and two paths are different if the sets of edges that they use are different. Run your algorithm to find the shortest path(s) between A-F, E-G and B-F.
- (b) (15 pts) Implement an efficient algorithm to compute **betweenness** of each edge in the graph  $G$  : the total number of pairs of nodes between  $s$  and  $t$  such that the edge  $(a, b)$  lies on the shortest path. In another words, it is the number pairs of nodes which has this edge in their shortest paths. For example, in Figure 1 the edge  $(B, D)$  has the highest betweenness. This edge is on every shortest path between any of A, B, and C to any of D, E, F , and G. Its betweenness is therefore  $3 * 4 = 12$ .

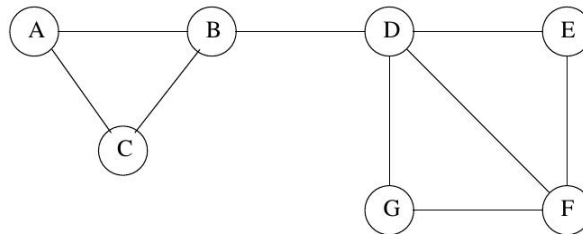


FIGURE 1 – Undirected Graph (input1.txt),  $G = (V, E)$

- (c) (30 pts) For a given **directed** graph in input2.txt,  $G = (V, E)$ , you should determine that the graph is **strongly connected** or not. A directed graph is *strongly connected* if there is a path between all pairs of vertices. If the given graph is strongly connected, your program must print "strongly connected" ; if not it must print "not strongly connected" to the screen. You may refer to some algorithms discussed in the class.

Report - 40 pts

- [10 pts] How do you find the shortest path(s) between two nodes? How do you compute betweenness of each edge? Show the betweenness value of each edge on the graph. How do you test if the graph is strongly connected or not? Explain in details.
- [15 pts] How does your algorithm work? Write your pseudo codes of a,b,c. Explain your classes and your methods. What are their purposes?
- [15 pts] Show complexities of your algorithms on pseudo-code.

*Submission*

You should be aware that the Ninova system clock may not be synchronized with your computer, watch, or cell phone. Do not e-mail the teaching assistant or the instructors your submission after the Ninova site submission has closed. If you have submitted to Ninova once and want to make any changes to your report, you should do it before the Ninova submission system closes. **Your changes will not be accepted by e-mail.** Connectivity problems to the Internet or to Ninova in the last few minutes are not valid excuses for being unable to submit. **You should not risk leaving your submission to the last few minutes.** After uploading to Ninova, check to make sure that your project appears there.

**Policy :** You may discuss the problem addressed by the project at an abstract level with your classmates, but you should not share or copy code from your classmates or from the Internet. **You should submit your own, individual project.** Plagiarism and any other forms of cheating will have serious consequences, including failing the course.

**Submission Instructions :** Please submit your homework through Ninova. Please zip and upload all your files using filename HW1\_studentID.rar. In the archived file, you must include your completed Report\_studentId file and all your program and header files.

All your code must be written in C++, and we must be able to compile and run on it on ITU's Linux Server (you can access it through SSH) using g++. You should supply one yourStudentID.cpp file that calls necessary routines for all questions (Multiple files are acceptable, as long as you state the compilation instructions in your report).

When you write your code, try to follow an object-oriented methodology with well-chosen variable, method, and class names and comments where necessary. **Your code must compile without any errors; otherwise, you may get a grade of zero on the assignment.**

If a question is not clear, please let the teaching assistant know by email ([kivrakh@itu.edu.tr](mailto:kivrakh@itu.edu.tr))