Ann, Bob, Charlie (*replace with your names*)
COSC 336
11/10/2020 (*replace with the current date*)

Assignment 10

Instructions.

1. Due Dec 2.
2. This is a team assignment. Work in teams of 3-4 students. Submit one assignment per team, with the names of all students making the team.
3. Your programs must be written in Java.
4. Write your programs neatly - imagine yourself grading your program and see if it is easy to read and understand. At the very beginning present your algorithm in plain English or in pseudo-code (or both). Comment your programs reasonably: there is no need to comment lines like "i++" but do include brief comments describing the main purpose of a specific block of lines.
5. You will submit on Blackboard 2 files. The first file should be a .pdf file with the solutions of Exercise 1 and Exercise 2, and also with a short description in English or in pseudocode of the algorithm for the programming task and the results that you are required to report. File 2 will contain the java codes of the program.

   For editing the above document with Latex, see the template posted on the course website.

   assignment-template.tex and

   assignment-template.pdf

   To append in the latex file a pdf file, place it in the same folder and then include them in the latex file with

   ```
   \includepdf[pages=-,pagecommand={},width=\textwidth]{file.pdf}
   ```

   To append in the latex file a .jpg file (for a photo), use

   ```
   \includegraphics[width=\linewidth]{file.jpg}
   ```

Exercise 1. Exercise 22.3 -2, page 610 in the textbook.

Notes:

• Read first the version of the DFS algorithm in the textbook on page 604 (it is also in Notes 10- graph traversals). This DFS traversal for directed graphs (called DFS with timing) assigns two numbers to each vertex: d (which is the discovery time) and f (which is called the finishing time). For the classification of edges you can use the information in Exercise 22.3.-5 (you don't have to solve this exercise, just use the given relations).

• Take into account the stipulations about processing vertices in alphabetical order (when you have to choose between two vertices). Otherwise I cannot grade your solution. Draw the graph (do it by hand if you have a hard time using tikz for this) and the d and f numbers next to each vertex. For the classification of all edges, make a table with two columns in which in the first column you write the edge and in the second column you indicate the type of the edge (tree, forward, back, or cross).

Exercise 2. Describe an algorithm that computes the shortest path from a given node s (called the *source*) to all the other nodes in an undirected graph G in which the edges can have the weights 1 or 2. Your algorithm should have runtime $O(n+m)$. Hint: you can use an idea that I said is bad in Notes 11, but which actually works fine for this set of weights. Describe your algorithm in plain English in a clear and concise way.

Programming Task.

In a weighted directed graph $G = (V, E)$, the *eccentricity* of a vertex $v \in V$ is defined by

$$\mathrm{ecc}(v) = \max\{\delta(v, u) \mid u \in V\},$$

where $\delta(v, u)$ is the weight of a shortest path from v to u (in other words $\mathrm{ecc}(v)$ is the smallest distance to the vertex u that is the furthest from v).

Write a Java program that computes the eccentricity of every vertex. Describe the method you are using.

The graph is given in a file that has two lines. The first line has n (the number of nodes, which are labeled $0, 1, \ldots, n-1$) and the second line has a list of n^2 numbers $w_0, w_1, \ldots, w_{n^2-1}$, where w_k is the weight of the edge (i, j) for $i = k/n$ and $j = k \pmod{n}$ (if $w_k = -1$, then there is no edge from i to j). Your program should read the file, build the adjacency matrix representation of G, and then compute with your method the eccentricities of all vertices.

Test your program on the graphs given in the files indicated and report the eccentricities.

input-10.1

input-10.2