University of Texas at Dallas—Department of Computer Science
CS 6380 Distributed Computing—Spring 2021
Project 1 Description

This is a group project: Form your own group of two or three members. Use elearning's discussion forum.

You will develop a simple simulator that simulates a synchronous distributed system using multi threading. There are n+1 threads in the system: Each of the n processes will be simulated by one newly created thread and there is one master thread (which is the main thread when your program starts executing). The master thread will "inform" all threads when one round starts. Thus, each thread simulating one process, before it can begin round x, must wait for the master thread for a "go ahead" signal for round x. Clearly, the master thread can give the signal to start round x to the threads only if the master thread is sure that all the n threads (simulating n processes) have completed their previous round (round x-1).

Your simulation will simulate the LubyMIS algorithm. The code (algorithm) executed by all processes must be the same.

The input for this problem consists of line 1 containing n (the number of processes of the distributed system which is equal to the number of threads to be created), line 2 containing one array uid[n] of size n; the i^{th} element of this array gives the unique identifier of the i^{th} process or i^{th} thread and all ids are integers and lines 3 to n+2 representing the connectivity matrix (either 0 or 1) with each line representing the adjacency information for a single process. Thus, line i+2 has n numbers (each being either a zero or a one). If the jth component of line i+2 is 1, then link (i,j) exists and if it is a zero, then i and j are not neighbors (link (i,j) does not exist.)

All threads know the value of n and each thread knows its neighbors (but not their ids) or local conectivity. Use only temporary ids as given in the LubyMIS algorithm for deciding whether a process is part of MIS or not.

To summarize, the input file input.txt has the following:

n
id array; a one-dimensional array of size n
adjacency matrix (which is a symmetric n by n matrix) with each row in a single line by itself.

The main thread should wait for all threads to terminate and output the number of rounds needed for all threads (simulating all processes) to terminate. The main thread must output the MIS and verify that the MIS constructed is indeed correct. Think about how the threads communicate among themselves and to/from the main thread. The MIS algorithm cannot assume the existence of the main thread.

Upload one tar file containing your source code, a README file that tells us how to compile and run, one sample input file input.txt and the result of running your program on your sample input file. Your project must compile and run on UTD's Unix servers (cs1 or cs2).

Due date: March 25, 2021, 11:55 pm.