# University of Texas at Dallas—Department of Computer Science
# CS 6380.001 Distributed Computing—Spring 2019
# Project 1 Description

This is a group project: Form groups of two members.

You will develop a simple simulator that simulates a synchronous network using multi-threading. There are n+1 threads in the system: Each of the n processes will be simulated by one thread and there is one master thread. 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 r to the threads only if
the master thread knows that all the n threads (simulating n processes) have completed their previous round (round r-1).

Your simulation will simulate the SynchBFS algorithm in synchronous networks. The code (algorithm) executed by all processes (threads) must be the same (except for the root process that we want the BFS tree to be rooted at).

The input for this problem consists of (1) n (the number of processes of the distributed system which is equal to the number of threads to be created) and (2) one array id[n] of size n; the i$^{th}$ element of this array gives the unique id of the i$^{th}$ process or i$^{th}$ thread, (3) which process is the root (a number between 1 and n), and (4) a connectivity matrix (a symmetric matrix) of n rows, with each row (containing n numbers and each number is either a 0 or a 1) providing connectivity information for one process. Thus, if n is 8 and the 5$^{th}$ row of this matrix is 0 1 1 0 1 0 0 0, then the 5$^{th}$ process is connected to processes 2, 3, 5 (of course it is connected to itself) by direct point to point links and not to the other processes by direct links.

The master thread reads input file input.dat containing these two inputs and then spawns n threads. The file input.dat is a text file and all process ids are positive integers. Process x can communicate directly with processes that are directly connected to it.

No process knows n. Each process knows its id, id of the root, and its local connectivity information when it starts. Thus, for example, 4$^{th}$ process knows its id, and the fourth row of the connectivity matrix and the id of the root.

Each process should know parent and all of its children before terminating.

Upload one tar file containing your source code, a README file that tells us how to compile and run, the sample input file (input.dat) and the result of running your program (script file output) on your sample input file.

Due date: February 14 2019, 11:55 pm.