Note: YOU MUST DO THE HOMEWORK BY YOURSELF. If you have difficulties in solving a question you may discuss it with friends, BUT you MUST phrase, write and formulate the answers by yourself, after you understand the solution. The language of the solution must be entirely your own.

Notice 2: Unless your hand writing is extremely clear, you should type (Word, or LaTeX, etc.) and print your homework.

Notice 3: Put your name and id number on each page of the solution.

1. In the first class (March 8) we saw a simple algorithm for Broadcast and Echo (in which one initiator sends a message to all other nodes in the network and receives a signal after all nodes are known to have received a copy of the message). In that algorithm we used one type of message, M (where M is the original message that is being disseminated).

(a) Here you are required to describe the algorithm again and provide its code (pseudo code) in such a way that two type of messages are being used, the message M, and ACK. In the this version, each node except the initiator, upon receiving M for the first time sends M to all its neighbors (including the neighbor from which it has received M in the first place). You are required to describe the code of the algorithm that results from the simple Broadcast algorithm combined with the Termination detection of diffusing computation. The complexity of the new algorithm should be very close to the complexity of the algorithm that was presented in class. What is the exact complexity (messages and time) of the new algorithm.

(b) Modify the code so at the end the root of the broadcast tree will know the depth (= height) of the tree (The longest distance in the tree from the root to any leaf node).

2. In class we saw an algorithm for Termination Detection of Diffusing computation. In that algorithm there were two types of messages, message type M (of the original diffusing computation) and message type ACK (FB) which was used to collect the acknowledgements in order to detect the termination of the diffusing process.

(a) Here you are required to describe the algorithm again and provide its code (pseudo code). What is the exact complexity (messages and time overhead over the diffusing computation) of the new algorithm as a function of \( m \) the number of messages sent by the diffusing computation.

(b) Modify the code so at the end the root node of the termination detection will know the total number of messages sent by the diffusing computation in the network.
3. Many distributed algorithms do not have a built-in termination detection mechanism. Given such an algorithm, we have to impose on it a termination detection mechanism, which will signal the nodes of the network that the algorithm has terminated.

Here is one such mechanism for non-FIFO model of computation, suggested by J.M. Helary, C. Jard, N. Plouzeau, and M. Raynal from IRISA France:

Every node $v$ maintains 2 matrices $SENT$ and $RECEIVED$. Entry $SENT(i, j)$ is the number of messages sent from $i$ to $j$ in the algorithm, to the best of $v$’s knowledge. Entry $RECEIVED(i, j)$ is the number of messages received by $j$ from $i$ in the algorithm, to the best of $v$’s knowledge.

When ever a node, $x$, becomes idle (from the underlying algorithm’s point of view) it floods the network with a message containing 2 vectors: $SENT(x, \star)$ and $RECEIVED(\star, x)$. Upon receiving a flood message every node updates its matrices accordingly.

Prove under the following assumptions, that if in some node $SENT(i, j) = RECEIVED(i, j)$ for all pairs $(i, j)$, then there is no message of the underlying algorithm in transit on any link of the network.

Assume a non-FIFO asynchronous network. Also assume that, magically all the $SENT$ matrices are initialized to the number of messages sent by the spontaneously starting nodes when they start (i.e., the first messages generated by the nodes which wakeup spontaneously to start the underlying algorithm, hence we assume that all start nodes start at the same time). $RECEIVED$ matrices are initialized to 0.

4. (a) There are $n$ identical finite automata connected by $n$ unidirectional, asynchronous links to form a unidirectional ring, $n > 1$. The same automata works for any size ring. Exactly one automaton (one node) receives a WAKE signal from the environment. Design an algorithm (pseudo code of the finite automata) which informs the predecessor of the awakened automaton that it is the predecessor of the awakened automaton. Show that your algorithm has asymptotically optimal bit complexity (I.e., prove a lower bound on the total number of bits exchanged by any such solution between all the nodes).

(Hint: The lower bound should hold even for arbitrary identical processors in place of finite automata, and variable length messages, even if the processors know $n$. Can you relax these restrictions farther?)

(b) Give a tight upper and lower bound (I.e., pseudo-code and lower-bound) for this problem in the synchronous model.