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 19) 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 two type of messages, message type M (the original message that is being disseminated) and message type Ack (Acknowledgement) which is used to collect the acknowledgements in order to detect the termination of the broadcast process.

(a) Here you are required to describe the algorithm again and provide its code (pseudo code) in such a way that only ONE type of message is used. I.e., you can *NOT* use the Ack type or any other type than M. 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 $c$ the total number of leaf nodes in the created tree, and $h$ the height of the created tree. The message $M$ may now carry a variable argument.

2. Here we modify the flood algorithm presented in the first class as follows: The flood algorithm is the phase of the broadcast and echo algorithm where only message M is being forwarded down to the nodes (without the echo). We want to modify the code so at the end of the algorithm the tree that is defined by the flood (as we defined in class) is a BFS tree. I.e., shortest path spanning tree from the initiator to the rest of the network. To do that each message $M$ is being tagged with the number of hops its sender is from the initiator. That is messages are now $M(i)$, where $0 \leq i \leq n$. Each time a node receives a message $M(j)$ where $j$ is smaller than the minimum value it has received so far, that node sends again a new message $M(j+1)$ to all its neighbors (or to all its neighbors except the one from which it got $M(j)$).

(a) Provide the code of the new algorithm.

(b) How do you suggest to detect the termination of this new algorithm.

(c) What is the message complexity of the new algorithm.
3. Given a graph $G = (V,E)$, the radius $r(v)$ of a vertex $v$ is the maximum distance $\max_{v'} d(v, v')$ from $v$ to any vertex in the graph. Suppose that you have an anonymous asynchronous message-passing network with no failures whose topology is a tree.

1. Give an algorithm that allows each node in the network to compute its radius.

2. Safety: Prove using an invariant that any value computed by a node using your algorithm is in fact equal to its radius. (You should probably have an explicit invariant for this part.)

3. Liveness: Show that every node eventually computes its radius in your algorithm, and that the worst-case message complexity and time complexity are both within a constant factor of optimal for sufficiently large networks.

4. There are $n$ agents arranged in a ring (only can talk to neighbors on the ring) enumerated 0 through $n - 1$. All calculations are mod $n$.

   Initially, agent 0 has $n$ tokens. The agents take steps asynchronously, and whenever agent $i$ takes a step in a configuration where it has a token but agent $i + 1$ does not, agent $i$ gives one token to agent $i + 1$. If either agent $i + 1$ already has a token, or agent $i$ has none, nothing happens. We assume that a fairness condition guarantees that even though some agents are fast, and some are slow, each of them takes a step infinitely often.

   (a) Show that after some finite number of steps, every agent has exactly one token.

   (b) Suppose that we define a measure of time in the usual way by assigning each step the largest possible time consistent with the assumption that that no agent ever waits more than one time unit to take a step. Show the best asymptotic upper bound you can, as a function of $n$, on the time until every agent has one token.

   (c) Show the best asymptotic lower bound you can, as a function of $n$, on the worst-case time until every agent has one token.

5. Consider computing on an anonymous ring (all processors in the ring are identical, no unique ids). The computation of a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ proceed as follows: For $n$ bits input $X = x_1...x_n$, $n$ copies of $p$ are arranged in a ring, and we number the processors from 1 to $n$, consecutively around the ring. (The processors don’t know this numbering and do not know $n$.) Each bit $x_i$ is input to processor $i$ at the same time, and the computation begins. Eventually, each of the processors should output $f(X)$ and halt. The function is cyclic, i.e., $f(x_1, x_2, \ldots, x_n) = f(x_2, x_3, \ldots, x_n, x_1)$.

   Show that if there is a processor $p$ which computes $f$, then $f$ is a constant function. (That is, $f(X)$ is independent of the input $X$.)

6. Based on the centralized Depth First Search (DFS) procedure, describe a distributed DFS traversal algorithm for an asynchronous network. In a traversal algorithm a central node, called $root$, initiates a token which has to visit all the nodes of the network, one at a time. Argue that in order to visit all the nodes the token must traverse all the links of the network. (Links might be traversed more than one time)
(a) (Half a page) Verify that your algorithm uses $O(|E|)$ messages and $O(|E|)$ time, where $|E|$ is the total number of links in the network.

(b) (At most one page) Modify your algorithm to reduce its time complexity to $O(n)$, where $n$ is the total number of nodes in the network. (Hint: in the modification there must be times at which there are more than one message in transit in the network.)