22C: 166 Distributed Systems and Algorithms
Homework 4, Total points = 60

Assigned 11/8/12 due 11/15/12

Please submit typewritten solutions through ICON, preferably in the pdf format. Late assignments will not be accepted without prior approval. In the world of distributed algorithms, the burden of proving (or arguing) that your solution will work is on you, and not on the reader.

Question 1. (15 points) Consider the implementation of a reliable channel from a sender process S to a receiver process R using a sliding window protocol. These days, communication links are mostly reliable, so assume that messages are never lost. However, messages can reach the receiver out-of-order, and the receiver can buffer at most one message.

(a) Can you implement the protocol using bounded message sequence numbers?
(b) How will you modify the protocol if messages are never lost in transit?

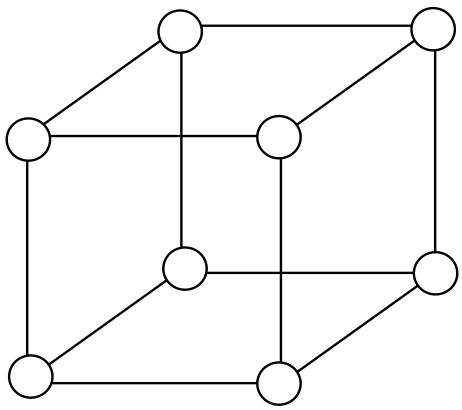
Question 2. (10 points) Construct an example to illustrate that in a synchronous system of seven processes, of which two can undergo byzantine failure, the OM(1) algorithm may not help the non-faulty processes reach consensus.

Question 3. (10 points) The byzantine generals algorithm helps reach a consensus when less than one-third of the processes undergo byzantine failure. However, it does not suggest how to diagnose such failures. Consider a system of five processes, of which *at most one process can be faulty*. Examine if the faulty process can be identified without any ambiguity. Investigate all relevant cases.

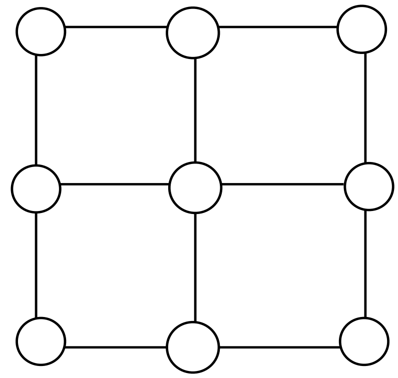
Question 4. (15 points) Propose a distributed algorithm to color the nodes of a tree (with no designated root) using exactly two colors 0 and 1, such that no two neighboring nodes have the same color. Assume that (1) each process has a unique identifier, (2) each process communicates with its neighbors using the shared memory model, and (3) actions are round-based (in each round every eligible process executes a local action). Estimate the time complexity of your algorithm in rounds.

Question 5. (10 points) In a spanning tree of a graph, there is exactly one path between any pair of nodes. If a spanning tree is used for broadcasting a message, and a process crashes, some nodes will not be able to receive the broadcast. Our goal is to improve the connectivity of the subgraph used for broadcast, *so that it can tolerate the crash of one process*.

What kind of minimal subgraph will you use for broadcasting, so that messages will reach every process even if one process fails? In each of the following two topologies, identify such a minimal subgraph.



(a)



(b)