Due: Oct. 13, 2 p.m.

Problem 1

Consider a group of distributed processors P1, P2, P3, and P4 that use the Ricarti-Agrawala algorithm for ensuring mutual exclusion. Assume that P4 is currently in the critical section and there is no other node in the WANTED state. Now consider requests from P1 and P2 (in that order) to enter the Critical Section.

1. Show the state and queue entries at each processor.
2. Now, P4 exits the CS and informs all relevant nodes that CS is released. Show the state and queue entries at each processor, at this stage.

Problem 2

In a certain system, each process typically uses a critical section many times before another process requires it. Explain why Ricart and Agrawala's multicast-based mutual exclusion algorithm is inefficient for this case, and describe how to improve its performance. Does your adaptation satisfy liveness condition in $ME2$? (Problem 15.7 in the $5th$ edition)

Problem 3

Is leader election possible in a synchronouse ring in which all but one processor have the same identifier? Either give an algorithm or prove an impossibility result.

Problem 4

Suggest how to adapt the bully algorithm to deal with temporary network partitions (slow communication) and slow processes. (Problem 15.9 in the $5th$ edition)

Problem 5

Modify the basic ring-based leader election algorithm to elect 2 leaders (two processes with the highest IDs).

Problem 6

Consider a synchronous system in which processors fail only by crash failure, with the additional constraint that the crash is always clean, that is, in a round, a processor either sends all its messages or none. Assume that in a round, a processor can send any number of messages, can receive any number of messages and process any number of messages.

1. For this system, design an algorithm that solves the consensus problem in the least possible number of rounds.
2. Explain why your algorithm is correct.