1. *5 marks* In a certain system, each process typically uses a critical section many times before another process requires it. Explain why Ricart and Agrawalaťs multicast-based mutual exclusion algorithm is inefficient for this case, and describe how to improve its performance.

2. *10 marks*

   (a) Construct a solution to reliable, totally ordered multicast in a synchronous system, using a reliable multicast and a solution to the consensus problem.

   (b) Construct a solution to consensus in an asynchronous system, using a reliable totally ordered multicast. Would it be possible to find a solution, on an asynchronous system with process failure, using a reliable but not totally ordered multicast and a majority function?

3. *5 marks* In a decentralized variant of the two-phase commit protocol the participants communicate directly with one another instead of indirectly via the coordinator. In phase 1, the coordinator sends its vote to all the participants. In phase 2, if the coordinatorŠs vote is NO, the participants just abort the transaction; if it is YES, each participant sends its vote to the coordinator and the other participants, each of which decides on the outcome according to the vote and carries it out. Calculate the number of messages and the number of rounds it takes. What are the advantages or disadvantages in comparison with the central variant?

4. *5 marks* Explain the difference between linearizability and sequential consistency, and why the later is more practical t o implement, in general.

5. *30 marks* The simplest algorithm to use in order broadcast in a network $G = (V, E)$ is the flooding algorithm. A convergecast phase can be added to the algorithm to produce an acknowledgement to the initiator when all processors of G have delivered the message that the initiator broadcasted.

   (a) Describe the flooding algorithm. Use both informal description and psudocode.

   (b) Describe the convergecast phase and how it composes with the flooding algorithm. Use both informal description and psudocode.

   (c) Prove the time and communication complexity of the flooding algorithm in a synchronous system.

   (d) Prove the time and communication complexity of the flooding algorithm in an asynchronous system.

   (e) Prove the time and communication complexity of the flooding-and-convergecast algorithm in a synchronous system.

   (f) Prove the time and communication complexity of the flooding-and-convergecast algorithm in an asynchronous system.

6. *5 marks* Describe an implementation of a distributed shared FIFO queue, shared by N processors. Your implementation should be able to support enqueue operation as long as not more than $\log N$ processors fail. Your implementation should use bounded space in the case where 50% of the operations are deques(enques), even in unbounded executions.