1. [10 points] Recall that in Raymond’s Algorithm for mutual exclusion, the processes are arranged in a logical tree. Consider a distributed system where the processes are arranged as shown in the graph below, which contains a cycle. The token is initially at process 1, and all holder pointers are initialized as shown in the figure. Does Raymond’s algorithm still satisfy both the safety and liveness properties in this system. If so, briefly explain why. If not, give a counterexample.

2. [10 points] Consider an asynchronous distributed system with three processes \( p_1, p_2, \) and \( p_3 \). The network is fully connected (every process can send messages to every other process), messaging is reliable, and all channels are FIFO. The processes circulate a token (message) around a logical ring, in a clockwise fashion, as shown in the figure below. Each process stores a single variable \( \text{count} \) that records the number of times it has possessed the token. Initially, every process has \( \text{count} = 0 \), and the token is at \( p_1 \). \( p_1 \) increments its \( \text{count} \) by 1 and sends the token to \( p_2 \). As soon as a process receives the token, it increments its \( \text{count} \) by 1, and then it sends the token to the next process in the ring.

Consider an execution of the Chandy-Lamport Distributed Snapshot Algorithm in this (fully connected) network. Suppose that \( p_1 \) has \( \text{count}=7 \), it sends the token to \( p_2 \), and then it immediately initiates the snapshot algorithm. Explain the execution of the algorithm, and give the recorded state of each process (its \( \text{count} \) variable), the location of the token, and the states of the channels.

3. [10 points] Dr. Science proposes the following modified version of the Diffusion Algorithm for Reliable Broadcast to reduce message complexity. Let there be \( N = 5 \) processes in the system:

**\( R_{bcast}(m) \):**
- Tag \( m \) with \( \text{senderID} \) and \( \text{seqNo} \)
- Send \( m \) to \( \lfloor \frac{N}{2} \rfloor + 1 \) other processes, chosen at random (excluding self)
- \( R_{deliver}(m) \)

**On \( \text{recv}(m) \):**
- If not delivered \( m \) already
  - Send \( m \) to \( \lfloor \frac{N}{2} \rfloor + 1 \) other processes, chosen at random (excluding self)
  - \( R_{deliver}(m) \)

Does Dr. Science’s modified algorithm implement Reliable Broadcast, i.e., does it always guarantee validity, agreement, and integrity? If so, explain why; if not, give a counterexample. Assume that the system model is the same as that discussed in class for Reliable Broadcast.
4. [10 points] Recall the Two-Phase Commit algorithm for synchronous systems with crash failures.

(a) Describe an execution of the algorithm in which all participants (including the coordinator) intend to vote commit, but the decision is to abort.

(b) Explain how this execution satisfies the agreement, validity, and termination properties.

5. [10 points] The FLP impossibility result states that there is no consensus algorithm that is totally correct in spite of one fault. Explain why the proof of this theorem applies only to asynchronous systems. Specifically, describe at least one part of Lemma 3 that would not be applicable in a synchronous system, and explain why it would not be applicable.

6. [10 points] Answer the following questions about the Synod Algorithm.

(a) In a system with 4 acceptors, is it possible for the acceptors (accNum, accVal) variables to simultaneously have the following values? If yes, describe briefly describe an algorithm execution that leads to this state. If not, explain why not.
   A1=(0,⊥), A2=(3, elk), A3=(1, llama), A4=(1, llama)

(b) In a system with 5 acceptors, what is the maximum number of different accVal values that may be present at the acceptors at any single time instance, not including ⊥?

7. [10 points] Consider an execution of the Byzantine Agreement Algorithm by Lamport, et al. in a system with 1 commander and 3 lieutenants. Assume the Byzantine Generals Problem can be solved in this system. Upon algorithm completion, each participant (loyal or not) outputs its decision, attack or retreat. A treacherous participant’s output cannot be trusted to reflect its actions in the algorithm. For each of the following, indicate whether it is a possible output of the algorithm.

(a) C=attack, L_1=attack, L_2=retreat, L_3=retreat

(b) C=retreat, L_1=attack, L_2=attack, L_3=attack

(c) C=retreat, L_1=attack, L_2=retreat, L_3=retreat

8. [10 points] Consider the specification of ZooKeeper that you were required to follow to build your replicated File System in Project 2 (as described in the slides). For each question below, answer yes or no. If the answer is yes, provide a justification. If the answer is no, provide a counterexample.

(a) Does the replicated file system guarantee linearizability?

(b) Does the replicated file system guarantee sequential consistency?

9. [10 points] A range query is a database operation that returns all data objects whose value is between a specified lower and upper bound. An example range query is a query for all objects with names that start with either the letter ‘J’ or the letter ‘K’. Consider a distributed storage system based on the Chord DHT, i.e., the object locations are determined by the Chord lookup function.

(a) Devise an algorithm for executing range queries on top of the Chord DHT. Your algorithm can call the lookup function as many times as needed.

(b) What is the message complexity of a single range query in your algorithm?

(c) Suggest a change to the Chord DHT that will improve the message complexity of range queries, and discuss the tradeoffs of this change in terms of the desirable properties of DHTs (load balancing, efficient lookups, and handing churn).