We'll post an outline of each recitation after the fact so you know what was covered in case you had to miss class that day. The recitation outlines are not a full replacement for recitation!

This week's recitation was largely a problem-solving session (good for exam prep). These are the problems we went over; some sections may not have gotten to every problem.

Question 1

In Paxos do proposal numbers have to be unique?

Question 2

A distributed system runs the Paxos algorithm with 5 acceptors (A1–A5) and 2 proposers (P1 and P2). The following events occur:

1. P1 sends prepare(10) to all acceptors.
2. A1, A2, and A3 reply with promises (no previously accepted proposals).
3. P1 sends accept(10, V1) to A1, A2, A3.
4. A1 and A2 accept (10, V1), but message A3 is dropped

Is value V1 accepted or is it still possible that another value gets accepted? If it is possible, what events would be needed?

Question 3

Consider a Paxos system with 3 acceptors: A1, A2, and A3, and 2 proposers: P1 and P2.
The following sequence of messages occurs:

1. P1 sends a Prepare(1) message to A1 and A2.
2. A1 and A2 respond with Promise(1) and report no previously accepted value.
3. P2 sends a Prepare(2) message to A2 and A3.
4. A2 and A3 respond with Promise(2) and report no previously accepted value.
5. P1 sends Accept(1, Value = X) to A1 and A2.
6. A1 accepts (1, X)
7. A2 rejects (since it already promised to proposal 2)
8. P2 sends Accept(2, Value = Y) to A2 and A3.
9. A2 accepts (2, Y)
10. A3 accepts (2, Y)

Question: Is this sequence valid? Which value (X or Y) will be chosen by Paxos?

Question 4

In a distributed system, the Paxos consensus algorithm is used with 5 acceptors (A1–A5). Two proposers, P1 and P2, are active in the system. Both continuously attempt to propose values:

- P1 proposes value V1 with proposal numbers 1, 3, 5, ...
- P2 proposes value V2 with proposal numbers 2, 4, 6, ...

Scenario

A. P1 sends a prepare(1) and receives promises from a majority.
B. Before P1 sends accept requests, P2 sends prepare(2) and also receives promises from a majority.
C. A majority rejects the accept requests from P1 as they already made a promise to P2
D. Before P2 sends accept requests, P1 sends prepare(3) and also receives promises from a majority.

and so on

Question: Can this happen with Simple Paxos as covered in class? If it is a valid sequence, what change to the protocol would you make to ensure that a value will eventually be chosen?

Question 5

We build a leader election system based on Paxos with three nodes, M1–M3, where each node acts as both an acceptor and a proposer. A leader is considered elected if a majority accepts the node's ID as the value in the latest Paxos instance.

Assume the network is initially functioning correctly, but at some point M1 becomes partitioned from the other nodes.

Is the state below possible? Is it still possible for M3 to learn that another node is the leader in instance I3? Will M1 or M3 still accept requests as the leader at the final stage as shown below?