Raft Practice Questions

Click next!

Adapted from Princeton COS 418 materials by Profs. Lloyd and Freedman

How to Read Commit Logs

	1	2	3	4
S0	1	1	2	3
S1	1	1	2	3
S2	1	1	2	3
S3	1	1		
S4	1	1		

In this diagram, there are five replicas, S0 - S4. Each box represents a log entry; the number/color tells the term when the log was added.

Here nodes S0, S1, and S2 have received logs from terms 2 and 3, but nodes S3 and S4 have not. Either they are very slow, or they are dead!

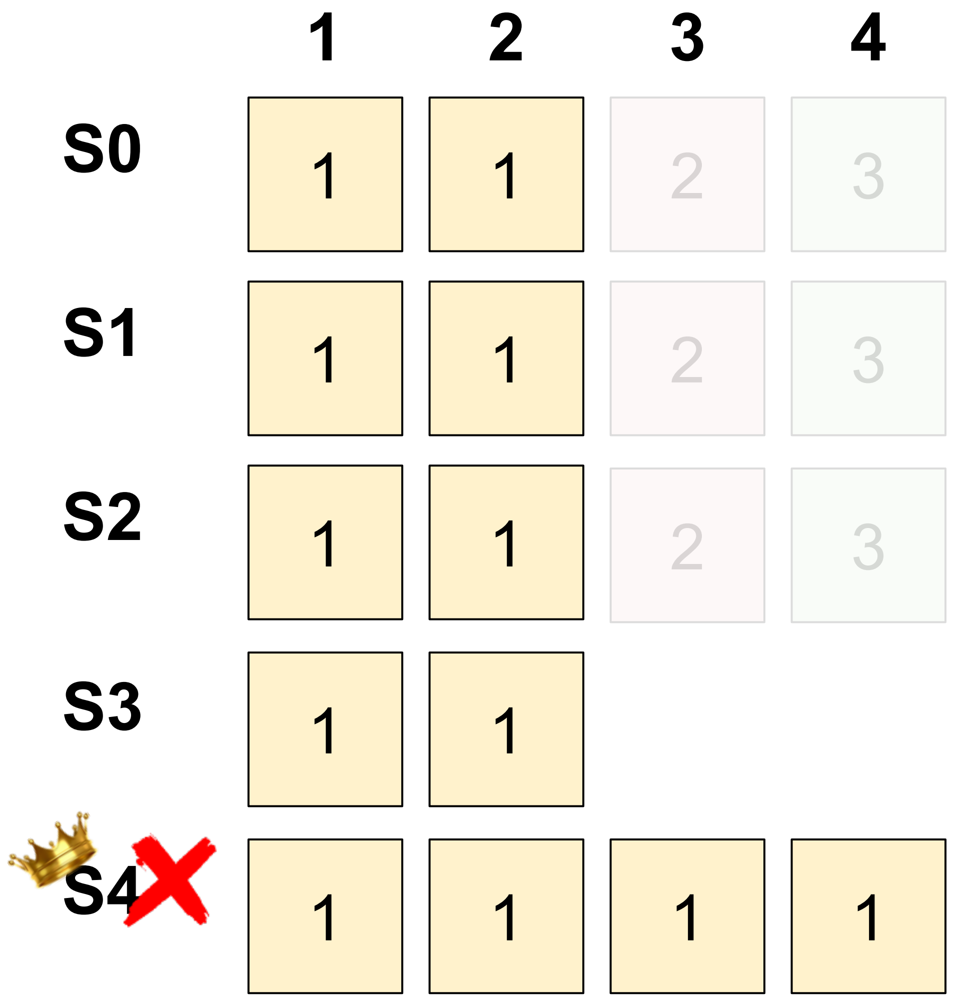
All logs in this diagram are committed since a majority of nodes have copies.

Question 1:

	1	2	3	4
S0	1	1	2	3
S1	1	1	2	3
S2	1	1	2	3
S3	1	1		
S4	1	1	1	1

Is this a possible configuration for the logs?

Answer 1a:



	1	2	3	4
S0	1	1	2	3
S1	1	1	2	3
S2	1	1	2	3
S3	1	1		
S4	1	1	1	1

Yes! This could happen if S4 used to be the leader for term 1. It must have accepted two additional logs before crashing.

A failed leader may have **uncommitted** logs from a prior term. Other nodes would roll back uncommitted logs if a new term starts.

(continued on next slide)

transparent entries haven't happened yet!

Answer 1b:

	1	2	3	4
S0	1	1	2	3
S1	1	1	2	3
S2	1	1	2	3
S3 ✗	1	1		
S4 ✗	1	1	1	1

After S4 failed, S0, S1, or S2 must have become leaders and added the logs for terms 2 and 3.

It appears that S2 also crashed. We don’t know exactly when, but it must have been before or during term 2.

Question 2:

	1	2	3	4
S0	1	1	2	3
S1	1	1	2	3
S2	1	1	2	3
S3	1	1	4	
S4	1	1	1	1

Is this a possible configuration for the logs?

Answer 2:

	1	2	3	4
S0	1	1	2	3
S1	1	1	2	3
S2	1	1	2	3
S3	1	1	4	
S4	1	1	1	1

No! If S3 recovers and starts a new election, no other node would vote for it!

A node will only vote for a node which has an **equivalent or more up-to-date** log!

- A node aware of a later term is more up-to-date
- A node with the same term, but more logs is more up-to-date

Question 3:

	1	2	3	4
S0	1	1		
S1	1	1	1	2
S2	1	1	2	

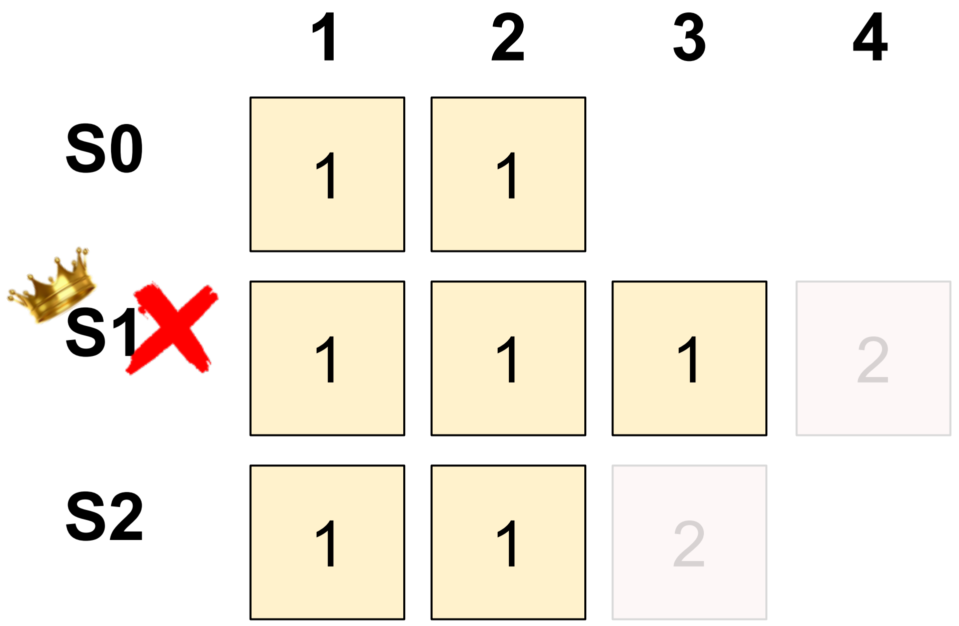
Is this a possible configuration for the logs?

Answer 3(a):

	1	2	3	4
S0	1	1		
S1	1	1	1	2
S2	1	1	2	

No! Let’s retrace the steps to understand why.

Answer 3(b):

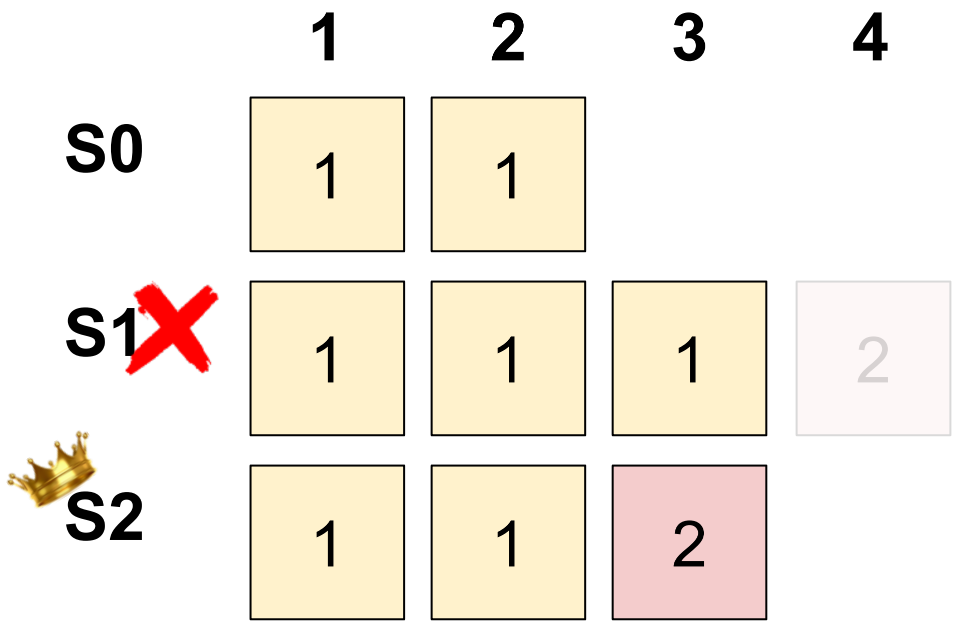


Previously we were in this state with just term 1 entries.

Since S1 has more logs, it must have been the leader.

Then it must have crashed.

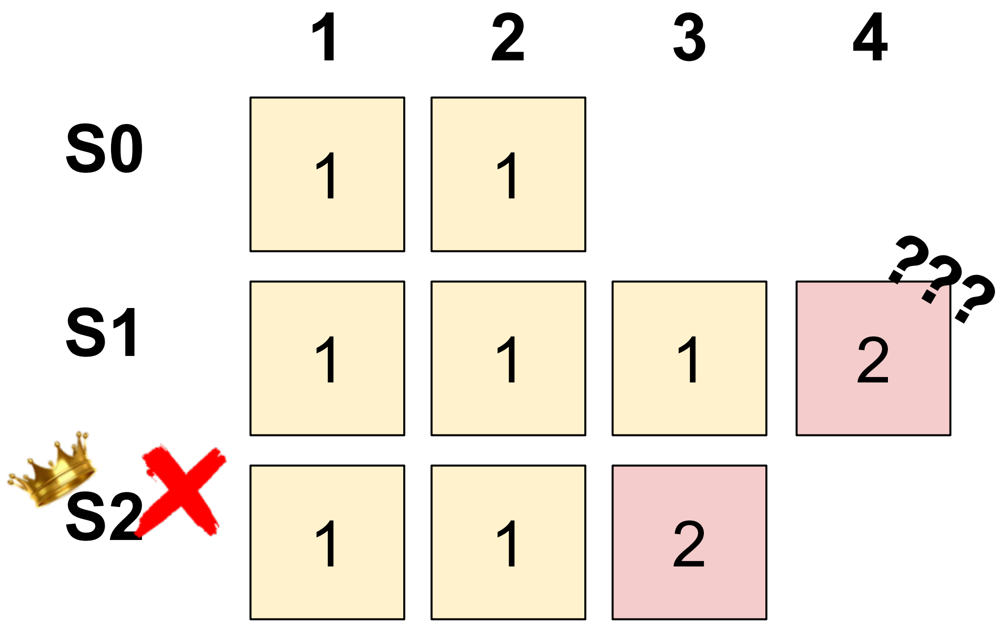
Answer 3(c):



After S1 failed, perhaps S2 won the election by getting votes from itself and from S0.

Then it added a log during term 2, but wasn’t able to replicate it.

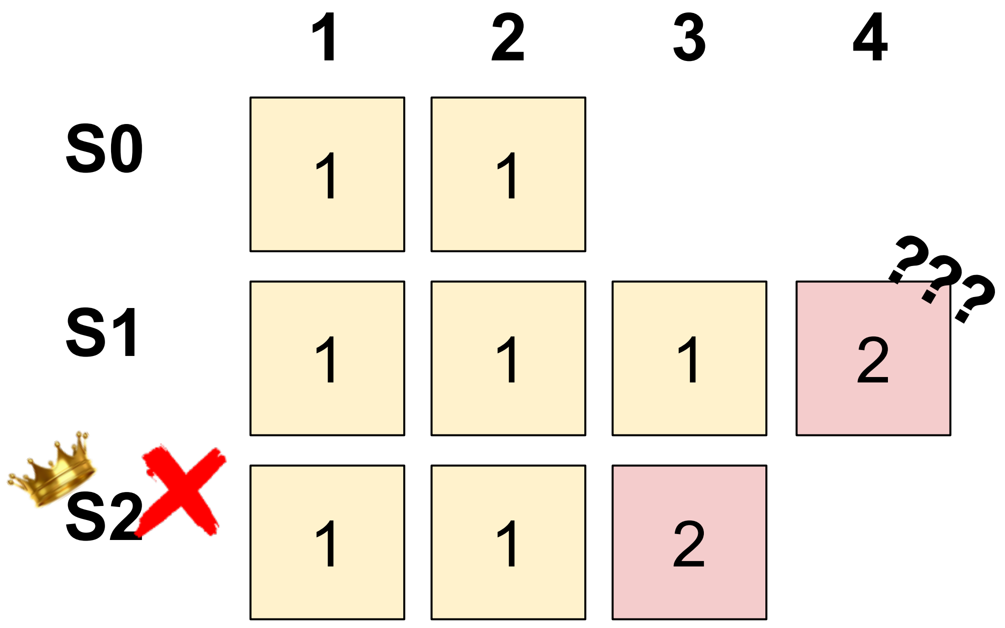
Answer 3(d):



If S2 then failed, we would need to run a new election. Since our assumption is only one node can be failed at a time, S1 must be back online.

Could S1 win the election?

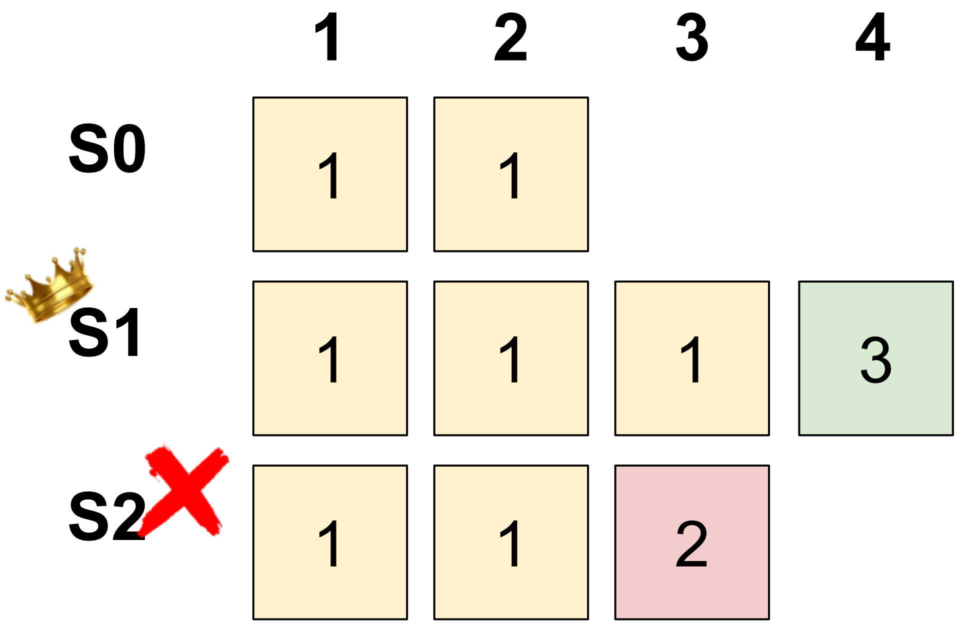
Answer 3(e):



Could S1 win the election?

No! S0 already voted in an election for term 2. Even though S0 didn't add a log for that term, it knows that the term started, and it will not vote for S1 if it proposes an election for term 2.

Answer 3(f):



The only way S1 could win the election is if it proposes to start term 3.

At this point, S0 and S1 can add logs for term 3. If S2 comes back online, it will have to discard its uncommitted log from term 2.

Nodes vote for the most up-to-date node and will never vote twice for the same election term.

A leader might have logs which need to be rolled back if it crashes and recovers later.