Consider the following illustration of the state of a Raft cluster.

1. Which node(s) can have been the leader for term 1? For each one, list the nodes that could have voted for them.

2. Which node(s) can have been the leader for term 3? For each one, list the nodes that could have voted for them.

3. Explain how it is possible that there is no entry for term 4.

4. We can classify the log entries in the figure into three categories depending on whether they will be committed to the state machines and thus returned to the clients: certainly, certainly not, and maybe. List for each of these categories the corresponding entries. Name entries by the node and the log index, e.g., a.1, a.2, a.3, and briefly justify the reason (you may of course group entries that have the same explanation).

5. Raft performs faster the smaller the election timeouts are. True or False, and why?
2 Gossip

1. Suppose a Gossip client sends the same query to two different replica managers. Will it necessarily get the same response from each? Explain.

2. Suppose two different Gossip clients send the same query to the same replica manager at approximately the same time (the replica manager receives no updates between its receptions of the two queries). Will the two clients necessarily receive the same results? Explain.


   (a) If this were done on Gossip using causal ordering for the updates, would client A be guaranteed that the value it reads for Y is either the same as that written earlier by B or causally dependent on that value? Explain. (Note: it’s not sufficient to say that Gossip guarantees causal ordering. You should either demonstrate that the value read by client A from server 4 satisfies the guarantee, or give a counter example. A good way to approach demonstrating this guarantee is satisfied or giving a counter example is to consider the vector timestamps of every client and server at each step in the process.)

   (b) Suppose we instead do the example on Bayou. Client A has various criteria it may use in choosing a suitable server (namely: read your writes, monotonic reads, writes follow reads, and monotonic writes). Are any of them sufficient for choosing server 4 so that the guarantee mentioned in part a holds? Explain. (You should give your answer in terms of the read sets, write sets, and DB mentioned in the lecture slides. When a client does a read, the DB of the server is added to the client’s read set. When a client does a write, the ID of the write is added to the client’s write set. During anti-entropy exchanges, the servers’ write sets are updated.)

3 Handing In

Once finished, you should hand in a PDF with your answers on Gradescope. Gradescope will allow you to select which pages contain your answers for each part of each question.

Please do not put your name on any page of your handin! This will allow us to do fully anonymized grading through Gradescope.

Please let us know if you find any mistakes, inconsistencies, or confusing language in this or any other CS138 document by filling out the anonymous feedback form:

http://cs.brown.edu/courses/cs138/s17/feedback.html