You are to work on the following questions *alone*. Do not discuss these questions with anyone. Typeset your answers and submit as a `PDF`.

1. In lab 4, ShardMaster and KVStore are separate client abstractions.

   (a) (5 points) List the client requests that are handled by each abstraction. (For example, the PaxosServer abstraction handles PaxosRequest.)

   (b) (5 points) In a sentence, explain why ShardStoreServers (despite being servers) are a client of the ShardMaster.

2. (10 points) In lab 4, suppose we have a client C and three Paxos groups, P, S, and T, each of size three (P.1, P.2, P.3, S.1, S.2, S.3, T.1, T.2, T.3). Each Paxos group has elected a leader (P.1, S.1, T.1). For a multi-key transaction involving all three groups and where P.1 is the coordinator, label which pairs of nodes directly communicate with each other. Please list each pair that communicate on a separate line, with the node earlier in the alphabet first, and omitting pairs that don't communicate. (For example, your first line will be C <-> P.1). Assume all messages are delivered in a timely manner and there are no failures.

3. Suppose we have two clients. Both clients perform the same multi-key transaction that transfers money from account A to account B (provided account A has sufficient funds). Code for this is given in the Spanner slides. Assume the two accounts are managed by different shards on different replica groups, and that each replica group acquires only the minimal lock needed for each operation (that is, a read lock for reads, a write lock for writes).

   (a) (5 points) Explain in a sentence or two how this might sometimes result in deadlock.

   (b) (5 points) Explain in a sentence or two how you could prevent or resolve this problem in your solution to lab 4.

   (c) (5 points) Suppose that one of the clients completed almost the entire transaction, including the updates to the two accounts, but failed just before sending the commit request. The client is not replicated and so will never complete. In a sentence or two, explain how would you implement transaction abort - to restore the state of the shards to the point before the transaction started.

4. (5 points) Give one scenario where GFS's "record append" would insert duplicate records at the end of a file.

5. (5 points) In two or three sentences, explain why using virtual nodes improves load balancing with consistent hashing.

6. (5 points) In a few sentences, explain why BigTable does not implement consistent hashing, and how despite this, it attempts to balance storage per server.

7. (5 points) Consider a cluster of 100 Dynamo nodes with the following parameters: $N = 3$, $W = 3$, $R = 3$. Dynamo picks as the write coordinator the node that responded fastest to the previous client read. In a sentence or two, explain why this system is not linearizable.