Your name: ____ **Your netid:** ________________

CS5412 Final Exam (May 4 version).

75m, with generous extra time for those who need longer or have SDS accommodations.
Closed book, no use of Internet. But you can use one page of home-made notes.

Q1. (20 pts). The motor vehicle databases for a set of states are being merged for the first time. You start by uploading the data into HDFS and then use Apache Hive to create a single sharded HBASE table that has one row for every violation. This very large table is sharded into **row groups** and **column groups**. You are designing a job to list drivers who have a total of more than 4 violations in any single 12-month period between 2015 and 2022.

a. **[5 points]** In one sentence each, what is HDFS? What is HBASE? How does HBASE relate to HDFS?

b. **[5 points]** What does it mean to say that HBASE is sharded, given that HDFS itself was already sharded? What is a column group? What is a row group?

c. **[5 points]** A MapReduce/Hadoop/Spark job can only access one chunk of data at a time in the map step, so for this table, map can only run on a single column group and row group at a time. But this means that infractions for a single driver might show up in multiple different chunks. How would you use MapReduce to collect the data for each individual driver?

d. **[5 points] Without writing any code**, just in words, tell us what LINQ is and how you could now design a LINQ expression to solve the problem.

Q2. (20 points). We learned that key-value systems such as CosmosDB and Cascade often have a versioned tuple replace in which you call **put** with some version number, and the tuple will be stored only if the version matches what you specified.

True or false (+5 points for a correct selection, -2.5 for an incorrect selection)

a. [T F] Using versioned **put** a set of tasks can update a shared (key,value) object such that every update will be performed exactly once, and will be totally ordered.

b. [T F] Compared to locking, the average delay for a versioned replace will be thousands of times less.

c. [T F] Applications written to use a versioned **put** need a loop.

d. [T F] Compared to locking, the versioned replace has simpler behavior if an application crashes while updating an object.

Q3. (20 points). FLP studies the problem of fault-tolerant consensus on a single bit, 0 or 1, with halting (crash) failures. In an asynchronous network that never loses a messages, the result shows that a correct protocol cannot be proved to terminate if even a single process might (or might not) crash.

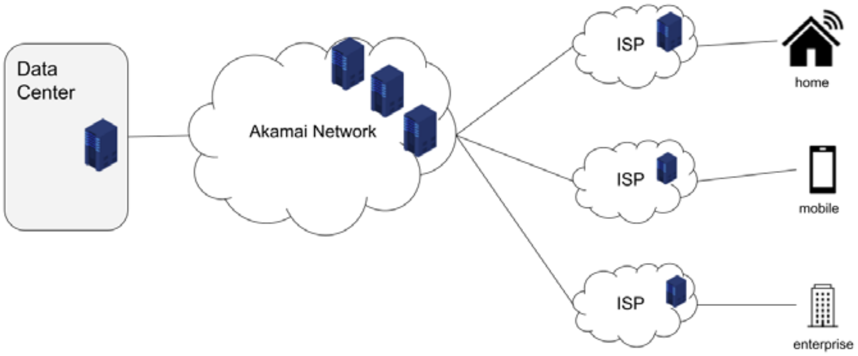
a. **[5 points]** Tell us what "correct" means in this situation.

b. **[5 points]** Tell us what "termination" means in this situation.

c. **[5 points]** Focusing on Paxos, we know that a write only has to reach a majority of the logs (the "acceptors"), and a read only needs to read enough logs to overlap with the writes. Does FLP apply to a program that uses Paxos as follows: the *n* processes all try and write their vote into the first Paxos log slot, and whichever one wins, we pick that value as the decision.

d. **[5 points]** Same scenario as part c. A criticism of FLP is that the attack it describes isn't very likely. If we take this into consideration, is it reasonable to say that Paxos is live after all?

Q4 (20 points). Anna Blasiak showed us this image of the Akamai content delivery network (CDN).



On the left is the data center of Akamai's customer (the "origin datacenter"). Akamai maintains its own network, seen in the middle, while also making agreements with Internet Service Providers (ISPs) who host Akamai CDN machines directly in the ISP's datacenters. The end-users are on the right.

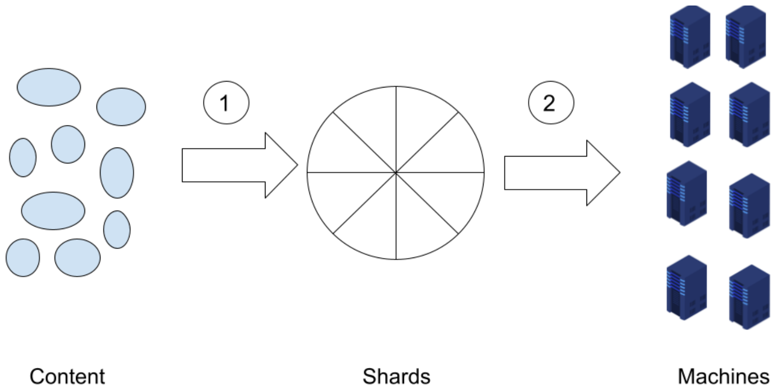
a. **[5 points]** Give two examples of the data that can be stored at an ISP's site.

b. **[5 points]** Explain how this design can benefit the end-users who access Akamai-based applications.

c. **[5 points]** Does this design have any benefits for Akamai or to the ISPs, as they operate this structure? Explain.

d. **[5 points]** What issues might arise with this ISP hosting approach that Akamai would not experience if it ran the entire structure itself, and didn't place Akamai machines in ISP data centers?

Q5. (20 points) Now let's take a closer look at the cloud datacenter. Inside the datacenter, Akamai utilizes sharding for their distributed data storage system. However, unlike the key-value examples we saw in other lectures, Akamai thinks of its sharding as occuring on a circular space of hashed keys, as seen in the illustration below.



In this graphic, the Akamai-hosted content is on the left. In step ① Akamai hashes the key for each content item, placing the key-value pair along the ring at the numerical location that the hashed value takes us to (the numbers wrap at the top). The ring is split into shards – our graphic shows eight shards. Step ② is managed by the team Anna works on, and involves storing the DHT into the servers, which are shown on the right. Select the statements that are correct.

True or False (+2 for each correct selection, -1 for each incorrect selection)

a. [T F] Because Akamai uses random hashing (like SHA 256) for step ①, and the number of key-value objects is enormous (billions), the mapping of content to shards should be relatively even, with roughly the same number of objects per shard. For example we would not see double the number of objects in one shard compared to the other shards.

b. [T F] Random hashing has a second benefit, which is that the expected request rate (load) for each shard should be quite balanced too. For example, we would not expect that a shard is somehow getting twice as many access requests as some other shard.

c. [T F] Akamai places the servers along the ring in a single round-robin manner: machine 0 handles shard 0, machine 1 handles shard 1, etc. (Note: C and D cannot both be true or both be false).

d. [T F] Akamai places the servers along the ring in a company-developed manner that involves running an algorithm that assigns each server to some "spot" on the ring. (Note: C and D cannot both be true)

e. [T F] When an object is placed into the ring, the *k* servers along the ring in clockwise order will host it and handle requests to it. *k* can vary for different objects based on how popular they are.

f. [T F] Because of E, a new server will need to receive a *state transfer* from some other server to initialize it. The state would be the set of objects it needs to have copies of, to host them.

g. [T F] Suppose that a server becomes unusually loaded. In this situation, Akamai can change where the servers reside in the mapping described in D, enabling them to give more load to a lightly loaded server or to spread load from an overloaded one.

h. [T F] If an image or video exists in multiple sizes, there will be a different cached object for each size of the object and Akamai will treat them independently.

i. [T F] As an additional caching feature, Akamai leverages caching on nearby mobile phones or laptops: if an application requests content that isn't available in Akamai, it will be fetched directly from a peer device first (one of the other systems on the extreme right in the first graphic). This ensures that content will only be requested from the originating data center if Akamai has never seen it before.

j. [T F] When a company uses Akamai CDN hosting, it still needs to be prepared for situations in which Akamai is temporarily not soaking up much (or any) load, so it might see some bursts of requests to its origin servers from time to time.

Q6. Extra credit, up to 5 points, counts only towards points lost on exams. Why did Microsoft create a new WiFi protocol, what was their key idea, and how did the resulting "white"-Fi protocol get used in FarmBeats?