

Sample Midterm Questions (& HW 04)

Below are some sample exam questions with which you can test your RAG “Cheat Sheet”.

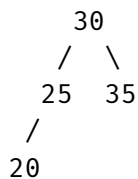
Question: What is the difference between a list where memory is contiguously allocated and a list where linked structures are used?

Answer: In a list where memory is contiguously allocated, there is no need for pointers. The next element in the list simply lies in the next block of memory. Therefore, a program can easily find the next element in the list by increasing the memory address by 1 unit. In a case where linked structures are used, the memory address of the next element in the list is not necessarily the next memory address like in contiguous allocation. Therefore, in linked structures, each element must have a pointer to the next element's memory address.

Question: When are linked lists faster than contiguously-allocated lists?

Answer: Linked lists are faster when you want to insert an element anywhere.

Question: Add 23 to the AVL Tree below. What imbalance case is created with inserting 23?



Answer: Since 23 is less than 30, we go to 30's left child (25). Since 25 is less than 25, we go to 25's left child (20). Since 23 is greater than 20, we place 23 as 20's right child. This presents a left-right insertion imbalance. 25 is the node of imbalance, and 23 was inserted into the right subtree of the left child of 25.

Question: Why is a B+ Tree a better than an AVL tree when indexing a large dataset?

Answer: B+ trees are the most optimal structure for indexing large datasets because they minimize disk access. B+ trees are “wider” than AVL trees, but this allows them to be shallower. The shallower a tree is, the less disk access we need to make to retrieve an element. Since each node in a B+ tree can contain an arbitrary number of sorted keys and their values, we can maximize the storage efficiency on disk.

Question: What is disk-based indexing and why is it important for database systems?

Answer: Disk-based indexing is a process by which we store frequently accessed data on disk. This allows for quick retrieval which is essential for database systems.

Question: In the context of a relational database system, what is a transaction?

Answer: In the context of a RDBMS, a transaction refers to a sequence of operations that are performed on data that are executed as a single, indivisible unit.

Question: Succinctly describe the four components of ACID compliant transactions.

Answer: Atomicity - each operation is a single, indivisible unit (like an atom). Consistency - operations take the database from one valid state to another, preserving the integrity constraints. Isolations - transactions execute independently without interfering with one another. If both transactions access the same data, they cannot affect each other's results. Durability - once a transaction is committed, its changes are permanent, even in the event of a system failure.

Question: Why does the CAP principle not make sense when applied to a single-node MongoDB

instance?

Answer: The CAP theorem only applies to distributed systems and a single-node MongoDB is just that - a single node. It is not a distributed system. Therefore, all three principles of the CAP theorem (Consistency, Availability, and Partition Tolerance) are fundamentally satisfied by a single MongoDB node.

Question: Describe the differences between horizontal and vertical scaling.

Answer: Horizontal scaling is adding more servers or nodes to distribute the load across more servers. Vertical scaling is making the current server/node larger to handle more information. The main difference in practice is that horizontal scaling is essentially limitless. You can add as many servers/nodes to achieve a more efficient load balance. Vertical scaling is more limited because you can only add so many CPUs or GBs of RAM for example.

Question: Briefly describe how a key/value store can be used as a feature store.

Answer: A feature store is a data system used for machine learning, serving as a centralized hub for storing, processing, and accessing commonly used features. Key-value stores can be used as a feature store because retrieving commonly used features is very efficient in key-value stores. Since the underlying data structure resembles a hash table, finding a particular feature can be done in constant time.

Question: When was Redis originally released?

Answer: 2009.

Question: In Redis, what is the difference between the INC and INCR commands?

Answer: INC is not a command in Redis. INCR increases a value by 1.

Question: What are the benefits of BSON over JSON in MongoDB?

Answer: MongoDB supports a wide variety of data types as its fields. Therefore, BSON is required to quickly serialize and deserialize complex field types such as documents. Using JSON may be slower in such complex cases.

Question: Write a Mongo query based on the movies data set that returns the titles of all movies released between 2010 and 2015 from the suspense genre?

Answer: `db.movies.find({ "year": { $gte: 2010, $lte: 2015 }, "genres": "suspense" }, { "name": 1 })`

Question: What does the `$nin` operator mean in a Mongo query?

Answer: Not in