DS4300 Midterm Prep - RAG Guide

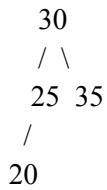
1. What is the difference between a list where memory is contiguously allocated and a list where linked structures are used?

- Contiguously allocated lists (like arrays) store elements in back-to-back memory locations.
    - Pros: Fast random access via indexing.
    - Cons: Insertions/deletions (especially in the middle) require shifting elements.
- Linked lists store elements as nodes connected by pointers.
    - Pros: Insertions/deletions are efficient if the node is known.
    - Cons: Slower access (no indexing) due to traversal.

2. When are linked lists faster than contiguously-allocated lists?

- When frequent insertions or deletions happen, especially at the beginning or middle of the list.
- When list size is unpredictable, so avoiding reallocation overhead is valuable.
- Random access isn't needed, and traversal-based access is sufficient.

3. Add 23 to the AVL Tree below. What imbalance case is created?



- Inserting 23 → goes to 30 → 25 → 20 → inserted as right child of 20.
- This causes a Left-Right (LR) Imbalance at node 25.
- Fix: Rotate left at 20, then right at 25.

4. Why is a B+ Tree better than an AVL tree for large datasets?

- B+ Trees:
    - Designed for disk-based access.
    - All data in leaf nodes — fast range queries.
    - Higher fan-out = shallower trees → fewer disk I/Os.
- AVL Trees:
    - Balanced in memory.
    - Not optimized for disk access or range scanning.

5. What is disk-based indexing and why is it important for databases?

- Storing indexes on disk instead of RAM.
- Used for very large datasets.
- Indexing structures like B-Trees minimize disk reads.
- Enables fast lookup, insertion, and deletion on persistent storage.

6. In a relational DB, what is a transaction?

- A transaction is a sequence of operations performed as a single unit.
- Either all operations succeed (commit) or none are applied (rollback).

7. Four Components of ACID Transactions

- A - Atomicity: All-or-nothing execution.
- C - Consistency: Maintains data integrity rules.
- I - Isolation: Transactions don't interfere with each other.
- D - Durability: Committed changes persist even during crashes.

8. Why does CAP not apply to single-node MongoDB?

- CAP Theorem applies to distributed systems.
- In a single-node MongoDB:
    - No partitioning possible.
    - So, the tradeoff between Consistency, Availability, and Partition Tolerance doesn't make sense.

9. Horizontal vs Vertical Scaling

- Horizontal Scaling: Add more machines (scale out).
    - Pros: More scalable, fault-tolerant.
- Vertical Scaling: Upgrade existing machine's hardware.
    - Pros: Simpler but limited by hardware.

10. How can a key-value store act as a feature store?

- Use entity IDs as keys and feature vectors as values.
- Fast retrieval of precomputed features during model inference.

11. When was Redis released?

- Redis was released in 2009.

12. Redis: INC vs INCR

- INCR: Increments the value at a key by 1.
- INC: Not a valid Redis command.

13. BSON vs JSON in MongoDB

- BSON:
  - Binary format.
  - Supports more types (e.g., Date, Binary).
  - More efficient for storage and transmission.
- JSON:
  - Text-based.
  - Easier to read, but limited in type support.

14. MongoDB Query: Suspense movies between 2010–2015

```
db.movies.find(
  {
    year: { $gte: 2010, $lte: 2015 },
    genre: "suspense"
  },
  {
    title: 1,
    _id: 0
  }
)
```

15. What does $nin mean in MongoDB?

- $nin: "Not in" — Matches values not in a given array.
- Example: { rating: { $nin: [3, 4] } } matches documents where rating is not 3 or 4.