Day 19 Study Notes 📝

1. Aptitude: Blood Relations

Blood relation problems test logical deduction regarding family ties. The most effective way to solve these is by drawing a **Family Tree**.

Key Symbols for Family Trees:

- **[ ] or Square:** Male
- **( ) or Circle:** Female
- **Horizontal Line (=):** Married Couple
- **Single Horizontal Line (-):** Siblings
- **Vertical Line (|):** Different Generations (Parent-Child)

Types of Problems:

1. **Dialogue/Introduction Based:** "Pointing to a photograph, a man said..."
2. **Relation Puzzle:** A complex set of relations involving multiple family members.
3. **Coded Relations:** $A + B$ means A is the father of B; $A - B$ means A is the sister of B.

2. Programming: Find the Majority Element

The goal is to find the element that appears more than $\lfloor n/2 \rfloor$ times in an array of size n.

Approaches:

- **Brute Force:** Nested loops to count occurrences of each element. $O(n^2)$ time.
- **Hash Map:** Store frequencies of each element. $O(n)$ time and $O(n)$ space.
- **Sorting:** The majority element must occupy the middle index if the array is sorted. $O(n \log n)$ time.
- **Boyer-Moore Voting Algorithm (Optimal):**
  - Initialize a `candidate` and a `count = 0`.
  - Traverse the array:
    - If `count == 0`, set `candidate = current_element`.
    - If `current_element == candidate`, increment `count`.
    - Else, decrement `count`.
  - **Time Complexity:** $O(n)$ | **Space Complexity:** $O(1)$.

3. Concept: Python Encapsulation

Encapsulation is the process of wrapping data (variables) and methods into a single unit (class) and restricting access to some components.

- **Public Members:** Accessible from outside the class. (e.g., `self.name`)
- **Protected Members:** Hinted by a single underscore `_`. Should not be accessed outside except by subclasses. (e.g., `self._salary`)
- **Private Members:** Hinted by double underscores `__`. Triggers name mangling to prevent accidental access. (e.g., `self.__atm_pin`)

4. C/C++ Concept: Access Specifiers

In C++, access specifiers define how the members (attributes and methods) of a class can be accessed.

Specifier	Within Class	Derived Class	Outside Class
public	Yes	Yes	Yes
protected	Yes	Yes	No
private	Yes	No	No

- **Data Hiding:** Private specifiers are the foundation of data hiding in C++, ensuring that internal object states cannot be corrupted by external code.

5. SQL: INNER JOIN

The `INNER JOIN` keyword selects records that have matching values in both tables.

Syntax:

```
SELECT columns
FROM table1
INNER JOIN table2
ON table1.common_column = table2.common_column;
```

Key Takeaways:

- **Intersection:** It behaves like a mathematical intersection. If a row in `table1` does not have a match in `table2`, that row will not appear in the result.
- **Efficiency:** Using `INNER JOIN` is generally more performant and readable than using a `WHERE` clause to link tables (Implicit Join).
- **Multiple Joins:** You can chain multiple `INNER JOIN` statements to link three or more tables together.