Day 21 Study Notes 📝

1. Aptitude: Direction Sense

Direction sense tests your ability to track movement through space and determine the final position or distance relative to the starting point.

Cardinal and Intermediate Directions:

- **Cardinal:** North (N), South (S), East (E), West (W).
- **Intermediate:** North-East (NE), North-West (NW), South-East (SE), South-West (SW).

Key Rules:

- **Right/Left Turns:** * Facing North: Right is East, Left is West.
  - Facing South: Right is West, Left is East.
  - Facing East: Right is South, Left is North.
  - Facing West: Right is North, Left is South.
- **Pythagoras Theorem:** Often used to find the shortest distance ("as the crow flies") between start and end points: $c^2 = a^2 + b^2$.
- **Shadow Problems:** In the morning (Sun in East), shadows fall to the West. In the evening (Sun in West), shadows fall to the East.

2. Programming: Check if a Matrix is Symmetric

A matrix is symmetric if it is equal to its transpose. Mathematically, for a matrix A, $A_{ij} = A_{ji}$ for all i, j.

Implementation Logic:

1. **Square Check:** A non-square matrix cannot be symmetric. Ensure rows $==$ columns.
2. **Comparison:** Iterate through the upper triangle (or lower triangle) and compare $A[i][j]$ with $A[j][i]$.
3. **Optimization:** You only need to check half the matrix (excluding the diagonal).

Complexity:

- **Time Complexity:** $O(n^2)$ because we must visit each element at least once.
- **Space Complexity:** $O(1)$ if checking in-place.

3. Concept: Python Generators

Generators are a simple way of creating iterators using functions and the `yield` keyword.

Key Features:

- **Lazy Evaluation:** They don't store the entire list in memory; they produce items on the fly only when requested.
- **Memory Efficiency:** Ideal for processing massive datasets where a standard list would cause a MemoryError.
- **State Retention:** The function "pauses" at `yield` and resumes exactly where it left off on the next call.

```
def my_generator(n):
    for i in range(n):
        yield i ** 2

gen = my_generator(5)
print(next(gen)) # 0
print(next(gen)) # 1
```

4. C/C++ Concept: Iterators

Iterators are objects that point to an element inside a container (like `vector`, `list`, or `set`) and allow traversal.

Benefits:

- **Abstraction:** They provide a uniform way to access different data structures without needing to know the underlying implementation (e.g., whether it's an array or a linked list).
- **STL Compatibility:** Most Standard Template Library (STL) algorithms (like `std::sort` or `std::find`) require iterators as arguments.
- **Safety:** They are generally safer and more flexible than raw pointer arithmetic.

5. SQL: FULL JOIN

The `FULL JOIN` (or `FULL OUTER JOIN`) returns all records when there is a match in either the left table or the right table.

Syntax:

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

Key Takeaways:

- **Union Effect:** It combines the results of both `LEFT JOIN` and `RIGHT JOIN`.

- **Handling NULLs:** If there is no match, the result set will contain `NULL` values for every column of the table that lacks a matching row.
- **Use Case:** Best used when you want to see all data from both sources and identify where the gaps (missing links) exist in both directions.