Task Breakdown

■ CPU-bound & Multiprocessing Tasks:

1. Matrix Multiplication (Pool vs. Single-threaded)

2. Matrix Multiplication with Process + Queue + psutil for CPU tracking

■Advanced Algorithm Tasks:

3. Min-Heap Task Scheduler

4. Segment Tree (Point Update & Range Query)

5. Dijkstra's Algorithm for Shortest Path (CLI-based)

📊 Expected Visual Outputs

- Speedup curves for matrix tasks (single-threaded vs Pool vs Process)

- CPU utilization plot (psutil) during matrix operations.

- Scheduler task timing histogram.

- Segment tree operation time plot.

- Pathfinding visualization using a graph or grid.

🧪 Report Highlights

- Performance benchmarks and comparisons.

- Clear rationale for choosing each concurrency approach.

- Explanation of data structure choices and complexity.

- Key Python concepts learned (GIL, multiprocessing, heapq, etc.).