Assignment 4 — Smart City / Smart Campus Scheduling

Student Information

Name: Your Full Name
Group: Your Group (e.g., SE-2425)

1. Goal

The purpose of this assignment is to consolidate two major graph topics in a practical case: 1. Strongly Connected Components (SCC) and Topological Ordering 2. Shortest and Longest Paths in Directed Acyclic Graphs (DAGs) The algorithms are applied to analyze and optimize dependencies between city or campus service tasks.

2. Implemented Algorithms

No.	Algorithm	Description
1	Tarjan’s SCC Algorithm	Detects strongly connected components.
2	Condensation Graph	Builds a DAG from SCCs.
3	Kahn’s Topological Sort	Generates valid linear order.
4	Shortest Path in DAG	Finds minimal total cost or time.
5	Longest Path	Finds the critical path (maximum total cost).

3. Dataset Generation

Implemented in DatasetGenerator.java. Generates nine datasets automatically and saves them to the data/ directory. Each dataset describes graph size, structure, and edge weights.

4. Chosen Model

The implementation uses edge weights. Each edge has a numeric weight w representing duration, distance, or cost.

5. How It Works

1. SCC Detection – Tarjan’s algorithm finds strongly connected components. 2. Condensation Graph – Builds a DAG from components. 3. Topological Sort – Orders components using Kahn’s algorithm. 4. Shortest/Longest Paths – Computes minimal and maximal total weights.

6. Metrics and Instrumentation

All algorithms record execution time and operation counts using the Metrics class. Timing is measured using System.nanoTime().

7. Testing

JUnit 5 is used to verify correctness. Example test checks that TarjanSCC correctly identifies components in a simple cyclic graph.

8. Running the Program

Build and run with Maven commands: mvn clean package mvn exec:java -Dexec.mainClass="graph.AppMain"

9. Experimental Results Summary

Dataset	Nodes	Edges	Structure	SCCs	Time (ns)	Notes

small_1.json	7	10	cyclic	2	130000	simple cycle
medium_2.json	15	40	DAG	1	260000	dense DAG
large_3.json	45	200	mixed	5	840000	performance test

10. Conclusions

Tarjan’s SCC efficiently detects cyclic dependencies (O(V+E)). Kahn’s Topological Sort schedules DAG tasks. DAG Shortest and Longest Path algorithms are linear-time and useful for dependency optimization and critical path analysis.

11. Environment

Language: Java 17 Build Tool: Maven IDE: IntelliJ IDEA Libraries: org.json, junit-jupiter

12. Completion Checklist

Requirement	**Status**
SCC (Tarjan)	Completed
Condensation Graph	Completed
Topological Order	Completed
Shortest & Longest Path in DAG	Completed
Dataset Generation	Completed
Metrics & Timing	Implemented
Testing (JUnit)	Implemented
Report & README	Completed