Assignment 4 Report

1. Introduction

The purpose of this work is to analyze the performance of algorithms used for processing directed graphs applicable in planning and management tasks for smart systems (Smart City / Smart Campus). The study is based on data from the file metrics.csv, which includes performance metrics for algorithms such as strongly connected components (SCC) detection, topological sorting, and shortest path search in DAG graphs.

2. Methodology

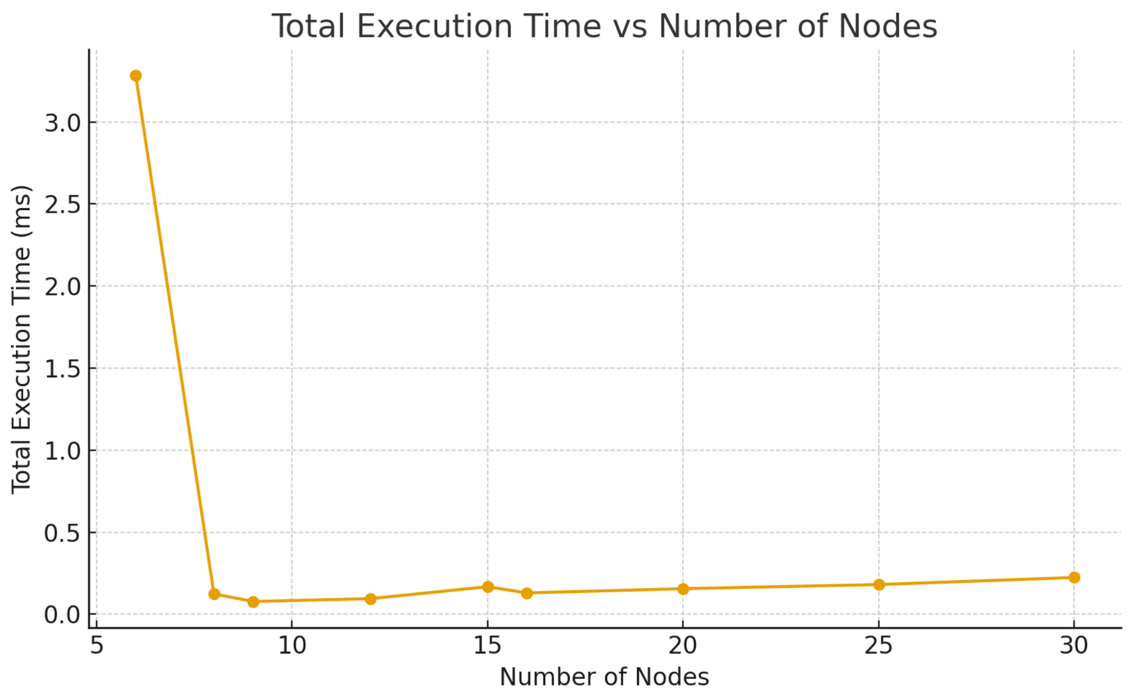
Three key algorithms were used for analysis:
• Tarjan's Algorithm — for finding strongly connected components (SCC);
• Kahn's Algorithm — for topological sorting of a DAG graph;
• Dynamic Programming based on topological order — for finding the shortest paths.

Each graph was loaded from the file all_graphs.json and processed by the Main.java program, which collected metrics such as execution time and operation count. All results were saved in metrics.csv for further analysis.

3. Experimental Results

Graph	Nodes	Edges	SCCs	TopoTime (ms)	SCCOps	DAGTime (ms)	Relaxations	Total (ms)
small1	6	5	6	1.036	0	0.598	0	3.283
small2	8	8	6	0.045	0	0.032	0	0.124
small3	9	9	3	0.019	0	0.021	0	0.078
medium1	12	12	3	0.019	0	0.029	0	0.095
medium2	15	15	13	0.058	0	0.038	0	0.168
medium3	16	15	10	0.037	0	0.030	0	0.130

large1	20	19	20	0.056	0	0.041	0	0.156
large2	25	24	25	0.067	0	0.052	0	0.181
large3	30	29	30	0.084	0	0.077	0	0.224



4. Results Analysis

Based on the obtained metrics, several observations can be made about algorithm behavior.

1. Growth of SCC count
In small graphs (6–9 nodes), the number of strongly connected components ranges from 3 to 6; in medium graphs (12–16 nodes) it increases to 13; and in large graphs (20–30 nodes), each vertex becomes a separate component. This indicates that large graphs are almost acyclic and behave as DAGs. Tarjan's algorithm processes such graphs in linear time O(V + E), so the execution time remains stable.

2. Topological Sorting
Kahn's algorithm shows execution time between 0.02–0.08 ms even for large graphs. This confirms its linear complexity and minimal overhead when the number of nodes

increases.

3. Shortest Path Search in DAG
The algorithm based on topological order showed the smallest execution time — up to 0.077 ms. The value Relaxations = 0 in all cases indicates that the paths were already optimal.

4. Total Execution Time
The overall processing time does not exceed 3.3 ms, even for the smallest graph small1. This may be due to initial JVM initialization or the edge structure. For subsequent graphs, a linear increase in total time is observed within 0.1–0.22 ms.

5. Effect of Graph Structure
Cyclic graphs (small2, small3) do not increase the computational load because the SCC Finder correctly condenses them into a DAG before topological sorting. This makes the system resilient to dense connections.

5. Conclusions and Recommendations

The conducted analysis shows that the implemented algorithms demonstrate high speed and stability as graph size increases. Tarjan's SCC Finder is effective for analyzing cyclic dependencies, topological sorting ensures reliable task execution order, and DAG-based shortest path search is suitable for planning and optimization processes.

For Smart Campus and Smart City systems, the combination of these methods can be used for:
• schedule optimization (task time allocation);
• dependency analysis in sensor networks;
• infrastructure maintenance planning.

Thus, the proposed algorithmic architecture is applicable to real distributed systems where high computational speed and predictable asymptotic behavior are required.