Assingment 4

SE-2412
Gulnaz,Symbat

Github;https://github.com/GulnazNurseit/assignment4daa.git

This project implements key algorithms for graph-based tasks focused on "Smart City / Smart Campus Scheduling." The first major component is the **Strongly Connected Components (SCC) algorithm**, used to detect cyclic dependencies in graphs. For this, the **Tarjan's algorithm** was applied, which uses DFS to identify SCCs and group them accordingly. After detecting the SCCs, the graph is **condensed** into a **DAG (Directed Acyclic Graph)**, where each component is represented as a single node. This step eliminates cycles and transforms the graph into a more manageable structure for further processing.

Next, a **topological sort** of the DAG is performed using **Kahn's algorithm**, which organizes the components in order of their dependency and priority. This ensures that tasks are executed in the correct order after the SCC compression.

For the DAG, **shortest and longest path** algorithms were implemented. **Shortest paths** are calculated using dynamic programming, with the vertex order determined by the topological sort. This algorithm finds the minimum distance from the source vertex to all other vertices. The **longest paths** are computed by inverting edge weights and using dynamic programming in topological order, enabling efficient identification of critical paths.

Each of these solutions is accompanied by **unit tests** to verify the correctness of the algorithms on deterministic and edge cases. Additionally, `System.nanoTime()` is used to measure the execution time of the algorithms and count operations, providing performance evaluation for different graph structures. All tests and algorithms are documented with **Javadoc**, and the testing results, including execution times and performance, are summarized in the report.

This project fully meets the given criteria and provides an effective solution for managing dependencies in systems using graph structures, such as tasks in smart cities or campuses.