Assignment 4 Report

SE-2412
Gulnaz,Symbat

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

Data Summary

Input Data Characteristics

- Graph Size: 8 vertices
- Graph Type: Directed graph
- Weight Model: Unweighted (all edges have implicit weight = 1)
- Source Node: 4 (for path calculations)
- Edge Structure: Contains cyclic components and linear chains

Results

Performance Metrics by Task

Task	Execution Time	Input Size (n)	Output Metrics	Status
Tarjan SCC	~55 ms	8 vertices	6 SCCs identified	✓ Passed
Graph Condensation	Included in topological test	6 SCCs	DAG with 6 nodes	✓ Success
Topological Sort	~50 ms	6 components	Valid linear ordering	✓ Passed
DAG Shortest Path	<5 ms	From source 4	Distance array computed	✓ Success
DAG Longest Path	<5 ms	From source 4	Distance array computed	✓ Success
Total Test Suite	105 ms	8 vertices	All algorithms	2/2 Tests Passed

Detailed Algorithm Outputs

Strongly Connected Components

SCCs: [[3, 2, 1], [0], [7], [6], [5], [4]]
- Total Components: 6
- Cyclic Component: [3, 2, 1] (3 nodes)
- Singleton Components: 5 individual nodes
- Component Distribution: One large SCC + multiple isolated nodes

Condensation and Topological Ordering

Topological order of components: [1, 5, 0, 4, 3, 2]
- DAG Size: 6 vertices (reduced from original 8)
- Order Validation: Valid linear ordering achieved
- Structure: Successfully eliminated cycles

Path Calculation Results

Shortest Paths from Source 4:
[2147483647, 2147483647, 2, 1, 0, 2147483647]
Longest Paths from Source 4:
[-2147483648, -2147483648, 2, 1, 0, -2147483648]
- Reachable Components: 2, 3, 4 (distances: 2, 1, 0)
- Unreachable Components: 0, 1, 5 (MAX/MIN values)
- Maximum Distance: 2 edges from source

Analysis

Algorithm Bottlenecks

Tarjan SCC
- Primary Bottleneck: DFS recursion depth
- Memory Usage: Stack allocation for recursion and state tracking
- Current Performance: Optimal for small graphs (55ms for n=8)
- Scalability Concern: Potential stack overflow for n > 10,000

Graph Condensation
- Processing Overhead: O(V+E) edge mapping between components
- Memory Usage: Temporary Set storage for duplicate elimination

- Efficiency: Minimal overhead in test case

Topological Sort (Kahn's Algorithm)
- Bottleneck: Queue operations and in-degree calculations
- Performance: Excellent (50ms for 6 components)
- Stability: Handles disconnected components gracefully

DAG Path Algorithms
- Efficiency: Single-pass topological processing
- Initialization: Array filling overhead negligible
- Constraint: Requires valid DAG input (ensured by condensation)

Structural Effects on Performance

Graph Density Impact
- Sparse Graph (current case): Fast processing, small SCCs
- Expected Dense Graph: Larger SCCs, more complex condensation
- Edge Distribution: Affects condensation complexity

SCC Size Distribution
- Current Structure: One 3-node SCC + five singleton SCCs
- Component Count: High (6 from 8 nodes) indicates sparse connectivity
- Cycle Impact: Single cycle minimally affects overall performance

Connectivity Analysis
- Source Reachability: Limited (3 of 6 components reachable from source 4)
- Path Distribution: Short maximum distance (2 edges) indicates localized connectivity
- Component Isolation: Multiple unreachable components

Recommendation: The current implementation is production-ready for small to medium graph analysis and provides a solid foundation for scaling to larger datasets with minor optimizations.