CSC 1204: Data Structures and Algorithms

Practical Assignment

Makerere University

March 11, 2025

Instructions

- You may work in a group of up to six members. No groups larger than six are permitted.
- All code must be hosted on a public repository (e.g., GitHub or GitLab). Include the repository link in your written report.
- The deadline for submitting the report (and repository link) is **March 25, 2025, 11:59 PM EAT**. Submissions after this time will be considered late.
- Each group will be required to present their report in class, demonstrating both their methodology and final results.

Task	**Marks**
1. Representation & Data Structures	5
2. Classical TSP Solution	10
3. SOM-Based Approach	10
4. Analysis & Comparison	10
Overall Clarity and Code Quality	5
Total	**40**

Note: The final 5 marks are based on code clarity, thoroughness, and the overall presentation of your work.

Traveling Salesman Problem Using Classical and SOM-Based Methods

Overview

This practical assignment requires you to:

- Represent and solve a small instance of the Traveling Salesman Problem (TSP) using a **classical algorithm** (e.g., Dynamic Programming, Branch-and-Bound, or Nearest Neighbor).
- Outline and implement a **Self-Organizing Map (SOM)** approach to approximate a solution to the same TSP instance.
- **Compare** the results (route distance, performance, complexity) between the classical algorithm and the SOM-based approach.

Use the graph shown in Figure 1 as your test data. It consists of 7 cities (nodes), with distances labeled on the edges. City 1 is the starting (and ending) city. All routes are bidirectional with the same cost in each direction.

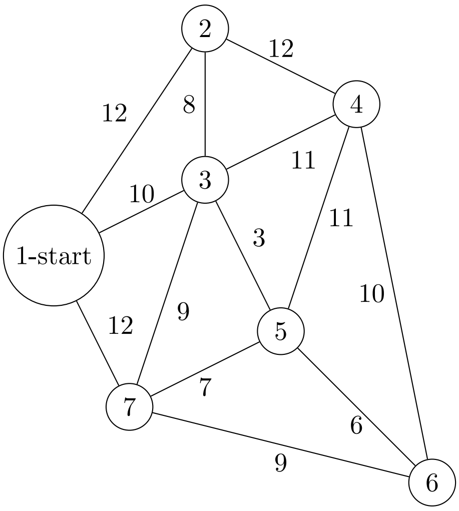

Figure 1: Graph representation of the cities and distances.

Tasks

1. **TSP Representation and Data Structures (5 Marks)**

   (a) *Graph Representation:* Show how you would store the graph in Figure 1 using an appropriate data structure. Briefly justify your choice by discussing how this structure supports efficient lookup of distances.

   (b) *Problem Setup:* Clearly restate the TSP objective: "Visit each city exactly once and return to the starting city while minimizing total travel distance." Confirm any assumptions made.

   Deliverables:

   - A short description of your chosen data structure (code snippet or pseudocode).
   - A brief explanation (1–2 paragraphs) justifying your choice of representation.

2. **Classical TSP Solution (10 Marks)**

(a) *Algorithm Selection:* Pick a classical TSP method (e.g., Dynamic Programming, Branch-and-Bound, or Nearest Neighbor).

(b) *Implementation:* Implement the chosen method in a programming language of your choice. Document your code with comments explaining key functions (e.g., state-space, recursion, bounding).

(c) *Results:* Output the final route and total distance. Share any intermediate results you find relevant.

Deliverables:

- Source code or well-structured pseudocode with comments.
- Final tour (sequence of visited cities) and total route cost.

3. **Self-Organizing Map (SOM) Approach (10 Marks)**

(a) *Conceptual Overview:* Briefly explain how an SOM can be adapted to solve TSP (initializing neurons, neighborhood function, learning rate, representing cities).

(b) *Implementation or Detailed Pseudocode:* Provide code or step-by-step pseudocode illustrating the core SOM-TSP logic (training loop, winner-takes-all update, decaying neighborhood, etc.).

(c) *Execution and Results:* Train the SOM on the given TSP graph data and present the final route and total distance.

(d) *Challenges:* Summarize any limitations or difficulties (e.g., parameter tuning, suboptimal convergence).

Deliverables:

- Written description (1–2 pages) of the SOM method for TSP.
- Either commented SOM code or clear pseudocode.
- The route found by the SOM and its approximate total distance.

4. **Analysis and Comparison (10 Marks)**

(a) *Route Quality:* Compare the routes obtained by the classical solution vs. the SOM approach. Indicate which is shorter (or if they match).

(b) *Complexity Discussion:* Outline the time complexity of your classical TSP method. Provide a high-level discussion of the computational cost of the SOM approach (number of iterations, updates per iteration, etc.).

(c) *Practical Considerations:* Discuss scenarios where an exact/near-exact solution is preferable versus using a heuristic like an SOM (consider number of cities, time constraints, memory usage).

(d) *Extensions:* Suggest at least one improvement or extension for the SOM or overall TSP approach (e.g., hybrid methods, alternative neighborhood functions, advanced heuristics).

Deliverables:

- A short report (1–2 pages) containing:
  - Comparison of route distances from both methods.
  - Time/complexity analysis of both approaches.
  - Discussion on trade-offs between classical and heuristic methods.
  - Suggestions for improvements or extensions.

End