# **Parallel & Distributed Computing Project**

***Parallel Optimization of Shortest Path Algorithms: Accelerating A-star, Bellman-Ford, and Dijkstra for Large-Scale Graphs***

## Task 1

### Group Members

1. Haseeb Ahmed , ERP: 26077
2. Abdullah Iqbal , ERP: 26904
3. Azizullah Khan , ERP: 24931

Code

## The attached code, developed for a **transportation company**, generates a dummy large-scale road network dataset and implements **Dijkstra, Bellman-Ford, and A-star algorithms** to compute the shortest paths between locations. The dataset simulates **a city's transportation system**, the code has a dataset generator which can be manually adjusted as per the requirements. The main code with all three search algorithms generates a random start node, target node and sweeps through the dataset to find the shortest path between them using the three methods. The output gives time duration taken by each method and the shortest path that is generated through each algorithm. This output allows us to compare the three algorithms in terms of correctness and time complexity. The code takes a lot of computational time as the size of the dataset increases which conflicts with practical scenarios where there is a huge dataset and retrieval from search space has to be done fast therefore our projects aim is to increase computational speed by implementing these fundamental and popular search algorithms in parallel contributing to efficient large-scale route optimization.

## Problem Statement

However, as observed in the code, Bellman-Ford takes significantly longer to compute shortest paths, which becomes a major concern as the transportation network expands. With an increasing number of routes and intersections, this prolonged processing time can cause delays in optimizing delivery schedules, making it essential to enhance computational efficiency. While Dijkstra & A-star provides faster results, they still face scalability challenges with extremely large datasets. As the network size grows, computation time follows a non-linear pattern, emphasizing the need for parallel computing to improve performance. This project aims to integrate parallel computing techniques using libraries such as OpenMP and CUDA-GPU concepts (whichever may work best) into the shortest path algorithms, optimizing route calculations and ensuring real-time efficiency for large-scale logistics and transportation systems.

## Base Papers

Our project is inspired by two key research papers. The first, *A Multithreaded-A star Algorithm for Optimizing the Graph Computation in Multicore Processing System*, motivated us to implement A-star in parallel to enhance computational efficiency and performance in real-world applications. The second paper, *Shortest Path Algorithms: A Comparative Analysis*, presents a variety of interesting algorithms, but we focus on two fundamental ones—Dijkstra’s and Bellman-Ford—analyzing their efficiency in parallel environment. By leveraging parallel processing, our research aims to optimize graph traversal and shortest path computations, making them more scalable and applicable to large-scale datasets. Through this study, we seek to improve execution time and resource utilization, contributing to advancements in parallel graph algorithms.

[Base Paper 1 Link](https://www.sciencedirect.com/science/article/pii/S1877050924005623?ref=pdf_download&fr=RR-8&rr=91b2b5116c0b21e4)

[Base Paper 2 Link](https://www.scielo.br/j/pope/a/H8zPR4pHnsSTgp6bNmPBWNr/?lang=en#:~:text=The%20shortest%20path%20algorithms%20were,previously%20mentioned%20in%20section%202.)

[Google Drive](https://drive.google.com/drive/folders/1jxaXPUdVEANpEkE041rnchiwASOaSUfR?usp=sharing)