

Programming Assignment 3: Implementation of Dijkstra with Naïve and Optimized algorithms

(Due: 31st Dec 2020, 11:59 pm)

In this final programming assignment, you will implement Dijkstra algorithm with the naïve algorithm and the optimized version using min-heaps and compare the performance of both the implementations. Dijkstra algorithm is a single-source shortest-path algorithm aimed to provide a shortest path from a given source to all the other nodes in the graph. Following are the details of the implementations.

Naïve Implementation:

Naïve implementation incurs a running time of $O(m * n)$ where m and n denote the number of vertices and edges respectively. This means that for a set of m vertices, each of the n corresponding edges are processed to determine the edge with minimum score.

Optimized Implementation:

The goal of optimized implementation is to perform better than the naïve implementation by leveraging a min-heap. In particular, Insert () and Extract-min () methods need to be implemented towards determining the edge with minimum score. Runtime of the optimized implementation would be $O(m * \log n)$.

Performance Comparison:

Finally, compare the execution time of naïve and optimized implementations for varying graph sizes and pictorially represent the execution time.

Input for the Graph:

The input for the graph should be from the file containing the following entries

<No of vertices> <No of edges> <source node>

<source node> <destination node> <cost>

...

Output:

<Source node> <Vertex 1> <Distance>

...

<Source node> <Vertex n> <Distance>

Submission Instructions:

Please submit the source code with the report containing the implementation of naïve and optimized algorithms along with the performance comparison on Teams before the due date.