EECS 630 Lab 03: Dijkstra’s algorithm

Objective

- Obtain a deeper understanding of greedy algorithms via the implementation of Dijkstra’s algorithm.

Specification

- Each row of the input files has the format of [ID_of_node_1 ID_of_node_2 Distance_between_nodes].
- Each row of the output files should have the format of [[IDs for the list of nodes contained in the shortest path] Path_length].
- Break ties by choosing the smaller node ID, if necessary.

Testing and Grading

We will test your implementation using the tester main function posted online. The posted input and output examples should be used for a testing purpose, while we will also use another set of inputs for grading. Your code will be compiled under Ubuntu 22.04 LTS using g++ version 11.4.0 (default) with C++11 standard.

Your final score will be determined by the success percentage of your program when fed with many random inputs. Note that if your code does not compile (together with our tester main function), you will receive 0. Therefore, it is very important that you ensure your implementation can be successfully compiled and pass the sample examples before submission.

For additional information, please read “README.txt” attached in the assignment package.

Submission and Deadline

Please submit your implementation as a single .h file, with a file name “MyDijkstra_[YourKUID].h”. For example, if my KU ID is c123z456, my submission will be a single file named “MyDijkstra_c124z456.h”. Submissions that do not comply with the naming specification will not be graded. All submissions will go through Canvas. The deadline is Fri Mar 29th, 2024, 11:59PM.