**Christian Marcellino**

**Data Structures and Algorithms II**

**Project 3**

**User's Manual**

**Setup and Compilation**

1. Download and unzip the submission from eLearning on a Linux box in the multi-platform lab.
2. The submission includes:
   * main.cpp
   * m.hpp
   * matrix.hpp
   * permutations.hpp
   * stopwatch.hpp
   * tsp-brute-force.cpp
   * tsp-brute-force.hpp
   * tsp-ga.cpp
   * tsp-ga.hpp
   * UserManual.docx
   * distances.txt
   * UMLDiagram.pdf
   * Results.xlsx
3. Environment: This program has been tested in the multi-platform lab and will run there.

1. Compiling. This program includes a Makefile. At the command line in Linux, type make. The program produces an executable entitled “main”

**Running the program**. Be sure that the distances.txt is in the same directory as the executable. Execute the ./main command. No arguments are required.

User input:

Four inputs are required and will show in the following order.

* + Number of cities to run (should be an integer 0 or greater)
  + Number of generations (should be an integer 0 or greater)
  + Number of tours in a generation (Should be an integer 0 or greater)
  + The percentage of a generation that should mutate (Should be a decimal between 0 and 1)

**Output:** All output goes to the console. Output will be like this:

Number of cities to run? (input should be an integer greater than 0): 10

Number of generations to run? (input should be an integer greater than 0): 10

Number of tours in a generation? (input should be an integer greater than 0): 10

What percentage of a generation should be comprised of mutations? (input should be a decimal number between 0 and 1): 0.5

Brute forcing a tour of 10 cities.

Path: 0 2 9 3 7 5 10 4 8 6 1 0

Optimal Total Cost: 404.68

Runtime for brute force: 291 ms

Estimating a tour of 10 cities.

Most Elite Path: 0 2 9 7 5 3 1 4 8 6 10 0

Total Cost: 544.62

Runtime for generations: 1 ms

Percent of optimal that the ga produced: 134.58%