**Justin Wymore**

**Data Structures and Algorithms II**

**Project 3**

**User’s Manual**

**Setup and Compilation**

1. Download and unzip DSA2\_Project3\_Wymore from eLearning onto a Linux platform.
2. The submission includes:
   1. Main.cpp
   2. BruteForce.cpp
   3. BruteForce.hpp
   4. Genetic.cpp
   5. Genetic.hpp
   6. Makefile
   7. Project3\_diagram.dia
   8. Distances.txt
3. Environment: This program has been tested in Ubuntu and Windows (mingw). For Linux, includes must be .hpp’s in main and/or TestCase. For Windows using mingw, headers must be .cpp in main and/or TestCase.
4. Compiling: This progam includes makefile. At the command line in Ubuntu or Command Line, type make clean main. The program creates an executable titled main.

**Running the program.**

1. distances.txt must be in the same directory as main.
   1. distances.txt must be of the format:
      1. “###.###” ‘\n’ or
      2. “###” ‘\n’
2. Issue the command ./main
   1. No command line arguments are required
3. You will be prompted to enter the number of cities. This number must be a positive integer less than or equal to 20.
4. You will be prompted to enter the number of tours that will be in each generation. This may be any positive integer, but it is recommended that it is significantly less than the factorial of the number of city’s entered.
5. You will be prompted to enter then number of generations that will be used. This may be any positive integer and recommended to be less than the factorial of the number of city’s entered
6. You will be prompted to enter the percentage of mutations you’d like to use. This should be an integer between 0 and 100

Output:

1. Output will be written to the terminal in the format:

The number of cities run was: ##

The optimal cost is: ###.##

It took ###.##### seconds to complete the Brute Force

The Genetic Algorithm found a cost of: ###.##

It took #.#### seconds to complete the Genetic Algorithm

And the percent of optimal is ###.###%