Sarah Stevenson

Data Structures and Algorithms II

Project 3

User’s Manual

Setup and Compilation

1. Download and unzip the submission from Canvas on a Linux box in the multi-platform lab.
2. This submission includes:
   * main.cpp
   * city.cpp
   * city.hpp
   * tour.cpp
   * tour.hpp
   * permutation.cpp
   * permutation.hpp
   * Makefile
   * UsersManual.doc (this file)
   * distances.txt (distance file for the array of city distances)
   * comparison (excel document showing comparison of 10-14 cities, with varying number of tours
3. Environment: This program has been tested in the multi-platform lab and will run there
4. 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 distances.txt is in the same directory as the executable. Issue the command ./spelled [A] [B] [C] [D] no command line arguments are required or checked.

[A] –integer of cities to be run

[B] – integer of tours to be run. Should be equal to or less than number of cities.

[C] – integer of number of generations to be run.

[D] – double of percentage of generations that are mutations. (ex. 50% will be read as .50)

Output: All output goes to the console. Output will be similar to this, assuming 4 cities, 4 tours, 2 generations, and .5 mutations. This is not an actual example, just example of format.

Note: “loading” is so user knows program is still running

Loading city 0/4

Loading city 1/4

Loading city 2/4

Loading city 3/4

City order: 0 1 2 3 0

Distance: 0.5

Brute time: 0s 8735394ns

Generation 1:

City order: 0 2 1 3 0

Distance: 1.0

Generation 2:

Permutation 1:

City order: 0 2 3 1 0

Distance: 0.75