Alex Reichel

Data Structures and Algorithms II

Project 03

User’s Manual

1. Setup and Compilation

Download and unzip Submission from e learning on. A Linux box in the multi-platform lab.

1. Submission includes:

* PermuationGenerator.hpp
* PermuationGenerator.cpp
* main.cpp
* GraphGenerator.cpp
* GraphGenerator.hpp
* GeneticTours.cpp
* GeneticTours.hpp
* BruteForce.hpp
* BruteForce.cpp
* Distances.txt

1. Environment: This program has been tested in the SSH server and Eclipse
2. Compiling: This program includes a Makefile, at the command line in Linux, type make. The program produces an executable entitled speller

Running the program: Be sure names.txt and Filemaker.cpp are in the same directory as the executable. Issue the command make or “ ./main\_print ” to run the program.

User will be asked to input the number of cities they would like to tour, then the user will be asked how many tours they would like to go on. The the user will then input the number of generations of tours the program will compile as well as ask the user for the percentage of mutations that should occur in the following generation.

Output: All output goes to the console. Output will display the total time taken by both solutions as well as their percent of optimal

Number of Cities to run

10

How many indiviudals tours are in a given generation

10

How many generations to run

10000

Percentage of a generation that should be comprimised of mutations

10

Implementing Genetic Algorithm

Best Tour was tour #99999 Costing: 475.71 Best Tour was: 0 10 5 13 3 8 1 15 17 14 0

Total time taken for 10 Cities: with 10 tours within 10000 Generations

17.321396

Implementing BruteForce Algorithm

Best Tour was tour # 174900 down below costing: 346.81

0 1 5 10 3 8 13 15 17 14 0

Total time taken for 10 Cities: 70.893215

Percent of Optimal that Genetic Algorithm Produced: 137.167%