

Benjamin Linam

Data Structures and Algorithms II

Project 3

Functional Decomposition

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.c
 - functions.c
 - functions.h
 - FunctionalDecomposition.txt
 - cityWeights.txt
 - makefile
 - Execution Times.docx
 - Users Manual for Project 3.docx (This document)
3. Environment: This program was designed and tested on Eclipse. It has also been tested in the Linux lab and does work as expected.
4. 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 cityWeights is in the same directory as the executable. While in the Linux Lab, navigate to the folder containing all of the files associated with Project2 and issue the command "./main". No command line arguments are required.

User input: When the user is prompted, the program will require the number of cities to run, the number of individual tours in a given generation, how many generations to run, and what percentage of a generation should be comprised of mutations.

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

```
CALCULATING OPTIMAL COST VIA BRUTE FORCE ALGORITHM. PLEASE WAIT.
```

```
The optimal cost determined by the brute force method for 10 cities = 133.47
Brute force algorithm took 0.0497872 seconds to complete
The best cost at 10 generations of 10 cities = 1533.32
Genetic algorithm took 0.0000238 seconds to complete
Percent of optimal = 8.70%
```