CS 224 Introduction to Python | Spring 2022

Programming Assignment 1

University of Wisconsin – La Crosse | *Due Date: February 24*

Lists and Files

Background: The *traveling salesperson problem* (TSP) is one of the most famous problems in computing. Simply stated, given a weighted graph, the goal is to find a path that begins and ends at the same vertex, visits all of the other vertices exactly once, and minimizes the total distance traveled (the sum of the weights on the edges traversed). This problem is a member of a class of problems called `NP-complete`. The defining characteristic of this class is that if a polynomial-time algorithm exists for any of the problems, then a polynomial-time algorithm exists for all of the problems in the class. Currently, no such solution has been found. On the other hand, no one has proven that no such algorithm can exist.

TSP is not simply of theoretical interest. This is a real-world problem that UPS and FedEx solve thousands of times every day. That's not quite true because finding optimal solutions to even a single sufficiently large TSP instance would take millions of years. Thus, UPS and FedEx find approximate solutions to the problem every day. This is done using an *approximation algorithm*. Approximation algorithms are used to find good, though not optimal, solutions to computationally expensive optimization problems (such as TSP). *Genetic algorithms* are approximation algorithms that work, generally speaking, according to the laws of Darwinian evolution. Through some number of generations, they refine initially very bad solutions into good ones.

Description: The data you will use for this programming assignment were generated by a genetic algorithm for the TSP using mapping data from the Google API. You will write a Python program that gathers data elements from each of a number of data files. It will then determine the minimum, maximum, and mean values for the data elements and report them. The data elements we want to evaluate represent the length of the shortest path found during a run of the algorithm as well as the first generation in the run of the GA at which that solution appeared/. Our goal, is to find the overall minimum, maximum, and mean for a large number of runs of the algorithm.

Details: The zipped file you downloaded includes this assignment writeup and a directory called `datafiles` that includes approximately 100 directories, each representing one run of the GA.. Each directory has a name of the form `run.xxx`, where `xxx` is an integer. You cannot depend on those integers being consecutive. Within each directory,

there are several files. The files we are interested in are `run.xxx.finalbest` and `run.xxx.genbest`. Each `finalbest` file contains a single line that appears as follows:

```
 G 2500 I 0 L 55 F 80469.000 1.887 (3735, 1024) 22 23 53 52 1 35 36 10 11
12 32 15 51 47 18 50 19 54 6 13 2 21 25 24 20 26 0 28 27 29 30 31 37 38 17
33 8 9 34 7 49 48 3 39 16 42 41 40 4 46 45 44 43 5 14
```

The value of interest in this file is the one after 'F', in this case 80469.000. It represents the length of the best solution found by that run of the GA. Each `genbest` file contains many lines, each for one generation of the GA. An example line looks like this:

```
Gen 1122 153086.392 87255.030 80469.000 22 23 53 52 1 47 35 36 10 11 12 32
15 51 18 50 19 54 6 13 2 25 21 24 20 26 0 28 27 29 30 31 37 38 17 33 8 9
34 7 49 3 48 39 16 42 41 40 4 46 45 44 43 5 14
```

The value of interest in this line is also 80469.000, the 5th element of the line. Your program should first read the length of the best solution from `run.xxx.finalbest`. It should then search `run.xxx.genbest` for the first generation at which a solution with that length was found. The path length and the first generation number are the data you need to track for each run of the GA.

Your program should produce concise, easy-to-read output that conveys all required information: number of runs represented, minimum value, maximum value, and mean value for each of the two data elements of interest.

Adhere to common coding conventions and **comment your code.**. Include a comment at the top that looks like this:

```
#
# CS 224 Spring 2022
# Programming Assignment 1
#
# This program performs simple analysis of a data element across a
# large number of data files representing  runs of a traveling salesman
# algorithm.
#
# Author: Your name here
# Date: February xx, 2022
#
```

Submission: The name of your program should be `analyze.py`. Submit your solution as a compressed directory by 11:59 PM on the due date. The name of the directory you compress must be: `LastnameFirstname-Prog01`. For example, my submission directory name, before zipping, would be `MathiasDavid-Prog01`.