Using the TSP student testing tool for the assignment

Your Traveling Salesman Problem (TSP) assignment comes with 3 files:

- SRN_TSP.py
- TSP_STUDENT_TESTING_TOOL.py
- TSP.csv

These files are designed to get you started on the assignment and help you test your code to ensure it is working when you submit.

Rename SRN_TSP.py

For the SRN_TSP.py you should replace the <SRN> with your student number, then edit the last 2 lines of the TSP_STUDENT_TESTING_TOOL.py file to point to the newly named file:

```
import SRN_TSP                                       # this is importing your file
testing_procedure(SRN_TSP)                           # this is calling the testing procedue with your file
```

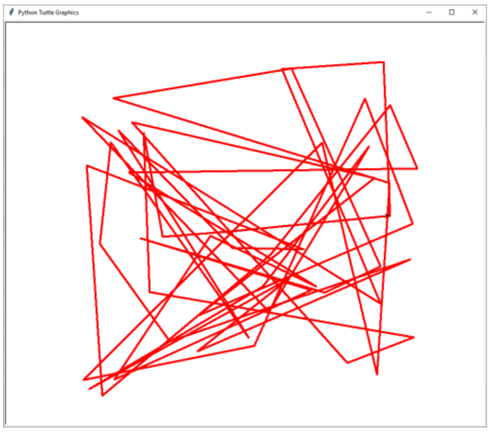
Aside from this you should not edit the TSP_STUDENT_TESTING_TOOL.py as this could compromise the result of the testing tool. When you submit your code will be passed through a similar tool so it is essential that you adhere to the provided structure.

The TSP.csv file?

The TSP.csv file contains a list of locations that are loaded into the testing tool when you run it. Do not edit the csv file, this is just to get you started, it is expected that your code will generate a map too, this is just here in case you have difficulty producing a map.

Running the testing tool

Initially when you run the testing tool (press F5 while in the testing tool file and make sure your student_TSP.py file is saved) you will notice it runs through 12 tests looking at each function. The tool will provide you with feedback within the shell as to what appears to be working and what does not, this is broken down by test. The tool is a very good indicator as to the result that you will get when the work is marked. The only exception to this is the print_map function that you will need to check manually by looking at the output, it should generate something similar to the image below.



Advice

Whilst the testing tool is useful for checking your functions meet the submission criteria, it is probably better and faster to call the functions within your own code while you are working on it, then test with the tool after. The reason for this is that whilst the testing tool gives you feedback, it is much more limited than the feedback generated by the shell with regards to errors.

Requirements for final submission

Ensure your final submission is named correctly and does not run anything when you hit run (no function calls within the file). Also do not include any print lines within the code as this will slow down the marking process.

Perfect feedback example

An example piece of feedback from a submission that works perfectly is as follows:

```
Test 1 - Student details function implementation
Function found successfully
Correct number of arguments found
Correct number of values returned
Student ID is using the correct data type (int)
Student username is using the correct data type (str)

Test 2 - Map generation function implementation
Function found successfully
Correct number of arguments found
Correct number of values returned
Correct data type for map generated
Map generated contained correct number of cities
Map generated coordinates within defined parameters
Map generated coordinates utilising full positive and negative range

Test 3 - Path calculation function implementation
Function found successfully
Correct number of arguments found
Correct number of values returned
Correct data type used to output path calculation

Test 4 - Print function implementation
Function found successfully
Correct number of arguments found
printing map function accessed

Test 5 - Nearest neighbour algorithm function implementation
Function found successfully
Correct number of arguments found
Correct number of values returned
Returning correct data type (list)
Nearest neighbour algorithm is correct, using all original provided locations

Test 6 - Create population function implementation
Function found successfully
Correct number of arguments found
Correct number of values returned
Gene pool contains correct number of individuals
Individuals in gene pool are correct and contain all locations
Individuals in gene pool are in a random order

Test 7 - Fitness function implementation
Accessing student create_population funtion to test this function
Function found successfully
Correct number of arguments found
Correct number of values returned
Gene pool contains correct number of individuals
Individuals in gene pool are correct and contain all locations

Test 8 - Mating function implementation
Accessing student create_population funtion to test this function
Function found successfully
Correct number of arguments found
Correct number of values returned
Gene pool contains correct number of individuals
Individuals in gene pool are correct and contain all locations
Your mating function appears to be working

Test 9 - Breeding function implementation
Accessing student create_population funtion to test this function
Function found successfully
Correct number of arguments found
Correct number of values returned
Individual output contains correct number of locations
Individual output is correct and contain all locations
Breed function is working correctly

Test 10 - Mutation function implementation
Function found successfully
Correct number of arguments found
Correct number of values returned
Individual output contains correct number of locations
Individual output is correct and contain all locations
Individual output has changed suggesting that some mutations have taken place
36 mutations have taken place with a mutation rate of 0.5

Test 11 - Iterator function implementation
Accessing student create_population funtion to test this function
Function found successfully
Correct number of arguments found
Correct number of values returned
Individual output contains correct number of locations
Individual output is correct and contain all locations

Test 12 - Full run of genetic algorithm with population of 5, 20 iterations, a mutation rate of 0.01 and an elite threshold of 0.1
Function found successfully
Correct number of arguments found
Correct number of values returned
Returning correct data type (list)
Genetic algorithm is correct, using all original provided locations
```

Results

Results from this assignment will be emailed to you, but there will also be an anonymous ranking table posted online to establish who’s algorithms performed the best both in terms of solutions found and the time required to get to that solution. This will add an element of competition so you can see how your algorithms compare to your peers, but don’t worry this will be anonymous.