COP4534 – Project 3

Objective

Students will be able to design, implement, and test solutions for known combinatorial optimization problems using concepts and tools studied in class. Students will address graphic visualizations of the problem solutions.

Assignment Problem

Design, implement, and test a solution for the following problem. Given a set of n points (x_0, y_0), (x_1, y_1), …, (x_{n-1}, y_{n-1}) in the plane, finding the shortest path that joins all of the points and starts and ends in the same point. This is a variation of the traveling salesman problem, in which the graph vertices are represented by the points in the plane and the weights on the edges are represented by the pairwise distances. shortest path between two vertices *s* and *t* of a weighted graph *G=<V, E>*. route that visits every vertex exactly once and returns to the starting vertex Consider the following requirements:

- The graph will be represented with an adjacency matrix, with distances as weights
- Input point coordinates will be integers, and weights will be positive real numbers
- The information of the data points will be provided in a text file
- A local search algorithm will be used in providing a solution to the problem
- Output, i.e., points and shortest route, will have a graphical nature. Display also the distance of the shortest output (in the window with the route and the points, not the console).

Please consult the class materials and the textbook, Section 12.6.2, for a discussion on local search.

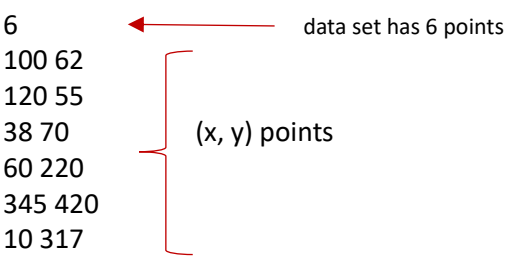
Extra Mile

Bonus points will be granted if the following is completed: in addition to the solution of the problem described above, find and graph a shortest path using exhaustive search. Both solutions, the one obtained with local search and the one obtained with exhaustive search, will be displayed in the same output window.

The input file will have this format:

- number of points
- list of the points, one point per line

For example,



Guidelines

The assignment is to be completed individually or in teams of two students. The given problem is based on the content studied in class on combinatorial optimization algorithms. You are allowed to use the code discussed in the lectures or in the textbook. In those cases, make sure you properly credit its source. Note that no installation of any software tool, package, or code should be required by anyone interested in running your code, other than a Java IDE and the Java™ Platform, Standard Edition Development Kit.

Deliverables:

- A compressed folder, *PID Project 3* (e.g. *1234567 Project 3*), containing

  - All of the source code of the exercise
  - Conclusions (Word or PDF file): what challenges did you find in implementing a solution this project? What did you learn?
  - Two screenshots of the running program (screenshots are *images*; they include, at least, part of the IDE and the code, and the complete graphical output). Screenshots will correspond to the solution of **two** different point sets, each with a minimum of 15 points.

- Include **only** the .java files mentioned above; do not include other files or folders generated by the IDE.

- Make sure you write your name(s), section, and Panther ID in the class comment section of each Java file.

Grading Rubric

The assignment is worth 150 points (out of 1000 total course points). Grade components:

<table>
<tr><th>Component</th><th>Points</th><th>Description</th></tr>
<tr><td>Submission</td><td>5</td><td>The student has submitted the project solution using the requirements for deliverables specified in the Deliverables section.</td></tr>
<tr><td>Organization</td><td>5</td><td>Code is expected to be neat, organized, and readable.</td></tr>
<tr><td>Content</td><td>140</td><td><table><tr><th>Deliverable</th><th>Points</th></tr><tr><td>source code</td><td>100 pts</td></tr><tr><td>conclusions</td><td>20 pts</td></tr><tr><td>screenshots</td><td>20 pts</td></tr></table></td></tr>
<tr><td>Extra Mile</td><td>60</td><td>Course extra credit. Note that points will be granted only if solution to Extra Mile section is complete; partial work will not be considered for a grade.</td></tr>
</table>