CM0889 Analysis of Algorithms
Programming Lab 1: Robot Tour Optimisation

Andrés Sicard-Ramírez

12th December 2020, version 1.1

1 General Information

The assignment is individual and you can use any programming language. The deadline is in the course web page.

2 Assignment (85%)

The textbook [Skiena 2012, § 1.1] introduces two heuristics and one solution for the Robot Tour Optimisation (RTO) problem:

A) the nearest-neighbour heuristic,

B) the closest-pair heuristic and

C) the exhaustive search algorithm.

The assignment for this programming lab is:

- (32.5%) To implement a program with the following specification:

  Input: A set P of n points in the plane with their coordinates. The input should be read from a file in the command-line arguments.

  Output: The shortest cycle tour that visits each point in the set P and the distance of this cycle according to A), B) *or* C).

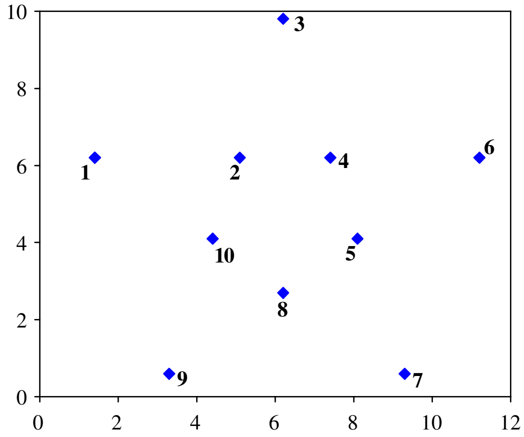
- (32.5%) Oral explanation of your solution.
- (10%) Testing your implementation.

  i) You should add at least *two successful* instances of RTO problem for which your heuristic/algorithm gives the shortest cycle tour.

  ii) If your heuristic fails in some instances of the RTO problem, that is, the computed shortest cycle tour is wrong, you also should add at least *two failing* instances.

- (10%) To document (in English) your source code. The documentation should explain your solution.

For simplifying your implementation you can suppose that the maximum number of points is 20.



city	x	y
1	1.4	6.2
2	5.1	6.2
3	6.2	9.8
4	7.4	6.2
5	8.1	4.1
6	11.2	6.2
7	9.3	0.6
8	6.2	2.7
9	3.3	0.6
10	4.4	4.1

Figure 1: Instance for the RTO problem.

2.1 Input Format

The input file is a text file. The first line contains the number of points n of the set P. The following lines contain for each point its coordinates x and y as *integer* numbers.

Example 2.1. The Figure 1 shows an instance for the RTO problem (figure from [Siqueira, Steiner and Scheer 2007, Fig. 1] where the problem is about cities instead of points).

Since coordinates x and y in the input file should integer numbers, we have the following input file:

```
$ cat p1.in
10
1 14 62
2 51 62
3 62 98
4 74 62
5 81 41
6 112 62
7 93 6
8 62 27
9 33 6
10 44 41
```

2.2 Output format

The expected output of your program on the instance in Example 2.1 is:

```
$ ./rto test/succeed/p1.in
Instance:
1 14 62
2 51 62
3 62 98
4 74 62
5 81 41
6 112 62
7 93 6
8 62 27
9 33 6
```

```
10 44 41
Solution: [ 10 9 8 7 6 5 4 3 2 1 10 ]
Distance: 371.59
```

Note that other solutions are also possible.

3 Requirements (10%)

Remark: Some requirements below are written for the C programming language. Adapt these requirements to the programming language of your choice.

i) The `main` function of your program should be in a file called `./src/rto.c`.

ii) Your program will be compiled using the following options:

`-Wall`,

`-Wextra`, and

`-Werror`.

iii) The three successful tests should be in a directory called `./test/succeed`.

iv) If necessary, the failing test should be in a directory called `./test/fail`.

v) Write a `Makefile` which at least the following variables and rules:

```
CC     := ...
CFLAGS := -Wall -Wextra -Werror

rto : rto.c

succeed-test : rto

fail-test : rto
```

4 Delivery (5%)

A. To create a *private* repository in GitHub. The name of the repository must be `cm0889-lab1`.

B. Share the repository with your lecturer (user @asr).

C. To add to the repository a `README.md` file in English containing at least the following information:

- Your name.
- Which did you implement? The nearest-neighbour heuristic, the closest-pair heuristic or the exhaustive search algorithm.
- Operating system version used.
- Programming language used.
- Compiler name and version used.
- Do not include any other files in the repository.

5 From the coordination

El control de versiones no es solamente un herramienta que facilitará la comunicación entre los miembros del grupo y la administración de los cambios al código. El control de versiones también ayudará al profesor y al monitor a llevar un control sobre el desarrollo de la práctica. Se espera que las diferentes registros dentro del control de versiones sean cambios graduales. En caso contrario, se procederá a realizar un escrutinio con el objetivo de evitar fraudes.

References

Siqueira, Paulo Henrique, Steiner, Maria Teresinha Arns and Scheer, Sérgio (2007). A New Approach to Solve the Traveling Salesman Problem. Neurocomputing 70.4–6, pp. 1013–1021. DOI: 10.1016/j.neucom.2006.03.013 (cit. on p. 2).

Skiena, Steven S. (2012). The Algorithm Design Manual. 2nd ed. Corrected printing. Springer. DOI: 10.1007/978-1-84800-070-4 (cit. on p. 1).