Advanced Data Structures and Algorithms

ESILV S7

Tutorial 9-10
2016 - 2017

All algorithms must be implemented in C or C++ language.

You have two sessions to complete this work.

It is important to analyse the problems and to conceive your algorithms before asking for your teacher's approval.

Just go allout in the code is always a bad strategy.

Deliver your workd on Moodle, at the latest at 23h59, the 4th day following the last session.

You must deliver an archive named : [ADSA-4A-GROUP-Name1-Name2].
Your deliverable presents itself in the form of an archive containing all the C++ or C source files (no executable) completed by a README.txt file containing a report on progress of your work and the main difficulties you encountered.

Assessment

The main assessment criterias are :

- Functional coverage, according to what has been asked.
  - Readability of the execution traces.
  - Ease of use and ergonomy of the interaction.

  For the executable
- For the code
  - Quality of the data structures and algorithms.
  - Quality of the code itself (indentations, comments, choice of identifiers, etc.)

— Thoroughness is also taken into account.
  — Respect of the delivery instructions, especially the deadline.
  — Quality of the README.

Work

The traveling salesman problem consists of a salesman and a set of cities. The salesman has to visit each one of the cities starting from a certain one (e.g. the hometown) and returning to the same city.
The challenge of the problem is that the traveling salesman wants to minimize the total length of the trip.

To do

We found in file named cites.txt a geographic latitude and longitude of center of few cities of the world.
We assume that : the edge exists between two cities if and only if the distance between the cities is greater than 100KM.

— Read data from the file.
— Load it in data structure representing the graph.

1. Find the maximum length path of the trip of traveling salesman problem starting from "PARIS" and visits each city exactly once and returning to "PARIS.
   (a) using an exact method to find an exact solution
   (b) using Lin-Kernighan heuristic
   (c) using Local search heuristic.
2. Estimate the complexity of those algorithms.
3. Implement an algorithm to find the minimum spanning tree. This spanning tree must contain the edge "PARIS —> SAINT GEORGES"

Each algorithm must be written in the file README.txt with explanation.