Homework 9: akka Actors

CSYE 7215: Spring 2019

The objective of this project is to develop an akka program that implements a solution to the following problem. Consider n cities at pair-wise distances represented by an array stored in a text file "cities.txt". The cities are numbered from 1 to n. Create 4 Searcher agents that are started concurrently and search for a path that starts at a given city S and ends at S while insuring that all cities are visited, and the length of the path is less than a given L. The user should be prompted to enter the values of S and L. Once an agent finds such a path, it sends the path and its length to all other agents and requests them to stop. All agents print their paths and a message that includes its ID and a statement – either "I won" or "<agent ID> won". The messages must be displayed in console and thus can be easily assessed for correctness by a human.

You will be given an example of the distance matrix, "cities.txt". But your program should work with any array of the same type. You will need to design an agent system that also includes additional agents (not just the 4 ones that perform the search), i.e., the User and the Solver agents.

In the development of this program you should use the Sequence Diagrams, as shown in the lecture slides. You can develop these diagrams in any drawing tool, e.g., PowerPoint or Visio. Or you can use a UML tool for that (all UML tools support creation of Sequence Diagrams). An example of a free UML tool is Star UML http://staruml.io/.

You can use any Java library functions publicly available to implement the search algorithm. Just provide the URL of the source in case you downloaded it from the Internet.

You can modify the files as you wish, provided they still implement akka. You should not change the name of the input file.

As usual, you should test your program for correctness. Among others, your program should check the input file, S and L whether they are "correct", e.g., the number of rows and columns is the same, and other preconditions and invariants.

You will be given an example skeleton files which you will need to adjust to your problem and then use for your project. You will also need to install akka on your system, as described in the lecture slides or in the akka documentation:

http://doc.akka.io/docs/akka/2.4/java.html#java-api

Submission

Submit a .zip file containing your project files to Assignments in Blackboard. The zip file you submit should have your name as the author. The zip file should also include a small document that will include the sequence diagram.

The file we gave you is just for your testing. We will test your programs on a different set of files. So please make sure that the program is not fixed on any specific file name; just let it read the file name as input. You can assume that that directory will contain those files.

To enforce academic integrity, your code may be checked for similarity to other submissions.