CMPT 340

Assignment 4 Due: Thursday, March 23, 2017, 11:59pm

Concurrent Programming and Actors

Total: 100 Points

Solve the following two problems using Java with Akka Actors.

Problem 1 (30 Points): *Fibonacci Numbers.* Develop an actor program for computing Fibonacci numbers concurrently. Particularly, create an actor which receives a request for a particular Fibonacci number from a client actor. If it is one of the base cases, the actor replies with a message containing the answer; otherwise, it creates the two sub-problems you would normally create in a recursive implementation. It then creates two new (Fibonacci) actors, and hands one subproblem to each. Once it has received replies from these actors, it adds them to produce its result, and sends it back to the computation's client.

Problem 2 (70 Points): *Stable Marriage Problem.* Let Man and Woman each be arrays of n processes. Each man ranks the women from 1 to n, and each woman ranks the men from 1 to n. A *pairing* is a one-to-one correspondence of men and women. A pairing is *stable* if, for two men m1 and m2 and their paired women w1 and w2, both of the following conditions are satisfied:

m1 ranks w1 higher than w2 or w2 ranks m2 higher than m1; and

m2 ranks w2 higher than w1 or w1 ranks m1 higher than m2.

Expressed differently, a pairing is unstable if a man and woman would both prefer each other to their current pair. A solution to the stable marriage problem is a set of n pairings, all of which are stable.

Write an Actors-based program using Java with Akka to simulate a solution to the stable marriage problem. The processes should communicate using asynchronous message passing. The men should propose and the women should listen. A woman has to accept the first proposal she gets, because a better one might not come along; however, she can dump the first man if she later gets a better proposal.

Submission:

Create a directory with your nsid as its name. Inside this directory, create separate sub-directories with names like problem2, problem3, etc. for each programming problem. Under each programming problem's folder, include a file with your program, as well as a text file showing a transcript of your testing of the program.

Once you have everything in your directory, create a zip file for the entire directory. If your nsid is <your_nsid> name the zip file <your_nsid>.zip. When opened, it should create a directory called <your_nsid>.

You may submit multiple times before the deadline, so you are advised not to wait till the last minute to submit your assignment to avoid system slowdown. You are encouraged to submit completed parts of the assignment early. Late submissions will not be accepted.