Problem Set 1, Part b

Due: Thursday, September 19, 2013

Problem sets will be collected in class. Please hand in each problem on a separate page. Students who agree to let us hand out their writeups can help us by writing elegant and concise solutions and formatting them using \LaTeX.

Readings:

Sections 3.6, 4.1-4.5 of Distributed Algorithms.

For next week: Section 5.1; Chapter 6; Aguilera, Toueg paper, listed in Handout 3; Keidar, Rajsbaum paper (skim).

Problems:

1. In this problem, consider comparison-based algorithms in bidirectional rings with UIDs.
   (a) Design an algorithm for the \textit{Mod-5 Counting Problem}, in which each process is required to output $n \mod 5$, where $n$ is the total number of processes in the ring. Prove an upper bound on the number of messages used in your algorithm. Try to get the smallest value you can for this measure.
   (b) Prove the best lower bound you can on the number of messages required to solve the \textit{Mod-5 Counting Problem} in the worst case.

2. (Based on Exercise 4.12, part (a).) Assume a network based on an undirected graph $G$, that is, a network in which communication is bidirectional between every pair of neighbors. Assume that the graph is connected, and that the processes have UIDs.
   (a) Describe informally an efficient algorithm that allows every process to output the exact number of edges in the network graph.
   (b) Give pseudocode in the style in the book for your algorithm.
   (c) Analyze the algorithm’s time and message complexity.

3. Consider a variation of the Shortest Paths problem described in Section 4.3 of the textbook with the following requirements:
   a. Processes are not required to halt. We require only that \textit{eventually} a shortest paths tree over the processes exists and does not change.
   b. Each process’s state contains a Boolean variable, \textit{source}, which is set to \textit{true} for $i_0$ and \textit{false} for every process $i \neq i_0$. It also has \textit{weight} variables, set to the weights of the incident edges; we assume the weights are positive integers.
   c. Except for \textit{source} and \textit{weight} variables, the initial state of each process is arbitrary: each non-source, non-weight state variable $v$ of a process, (including any round variables), is initially set to an arbitrary value of the right type.

   Note that these conditions imply that different processes might “think” that they are in different rounds, and processes can’t tell if they are just starting an algorithm execution or are in the middle of an execution.
(a) Explain informally why the BellmanFord algorithm described on p. 62 of the textbook does not solve this new version of the Shortest Paths problem.

(b) Describe informally a modified version of BellmanFord that does solve the Shortest Paths problem under these conditions.

(c) Give pseudocode in the style in the book for your new algorithm.

(d) Extra Credit: State an invariant which holds for your algorithm and when proved by induction implies its correctness (i.e. that it terminates and produces the correct output).