Problem Set 1, Part b

Due: Thursday, September 22, 2011
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; (Optional) Keidar, Rajsbaum paper.

Problems:

1. In this problem, consider comparison-based algorithms in bidirectional rings with UIDs.
   (a) Design an algorithm for the Mod-3 Counting Problem, in which each process is required to output \( n \mod 3 \), 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 Mod-3 Counting Problem in the worst case.

2. Exercise 4.11

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 eventually a shortest paths tree over the processes exists and does not change.
   B. Each process’s state contains a Boolean constant, source, which is set to true for \( i_0 \) and false for every process \( i \neq i_0 \).
   C. Except for source and edge weight constants (which are assumed to be positive integers), the initial state of each process is arbitrary: each non-source, non-weight state variable \( v \) of a process, (including any rounds or status variables), is initially set to an arbitrary value of the right type.

Note that these conditions mean that different processes might think 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 solves 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).
4. Recall that $\text{SynchGHS}$ proceeds in phases, each of which consists of a fixed number of rounds upon which processes agree ahead of time. We argued that this fixed number can be $O(n)$, where $n$ is the number of nodes in the network graph. Determine constants $a$ and $b$ such that every phase can be $an + b$ rounds (i.e., no phase requires more than $an + b$ rounds), and argue that $an + b$ rounds are sufficient for any phase. Try to find the smallest possible value for $a$ and $b$, particularly $a$. (Do not try to improve $\text{SynchGHS}$; just analyze the version described in the book.)