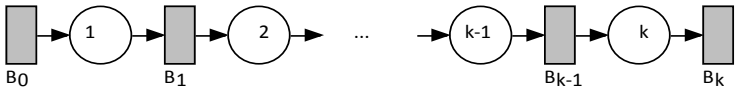
22C: 166 Distributed Systems and Algorithms
Homework 1, Total points = 60
Assigned 9/7/11 due 9/14/11

Please submit typewritten solutions through ICON, preferably in the pdf format. Late assignments will not be accepted without prior approval.

Question 1 (15 points): Consider the following linear pipeline that is a distributed system with k processes 1 through k:



Each process i can access two buffers: B_{i-1} and $B_{i.}$ These processes collectively compute the set of all primes within the set of the first n natural numbers. Initially buffer B_0 contains the integers 1 through n. In every round, each process i executes the following program:

do $\exists x \in B_{i-1}$ | x is unread and x is not divisible by i+1 → mark x as read; put x in $B_{i.}$ **od**

What is the *smallest number of rounds* needed to compute all the prime numbers? Briefly justify your calculation. It is you responsibility to choose an appropriate value of k so that the algorithm works.

Question 2 (15 points): Consider a completely connected network of n processes 0 through n-1. In this network, some process i wants to broadcast a message m to every other process using a synchronous message-passing model. In each round, a process can forward a message to exactly one other process. The goal is to complete the broadcast as quickly as possible.

Propose an algorithm using which the broadcast is completed in the *fewest number of rounds* (only the main idea using pseudo-codes is needed here). Calculate the time complexity in rounds.

Question 3 (10 points): Consider the clock phase synchronization algorithm discussed in the class (see page 78 of the textbook). We discussed that it works on a line topology. Can it be used on a ring topology? Briefly explain.

Question 4 (20 points): In a completely connected network of n *anonymous* processes, initially all processes are black. Design an algorithm so that *exactly k (0 < k <n) of them* turns red. Allow all processes to run their programs concurrently in lock-step synchrony, i.e. in each step every process eligible to execute an action does so. Using the locally shared memory model, propose an algorithm for doing this. Only the broad idea is needed (using clear English or pseudo-codes). [Hint: Use coin flipping]