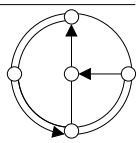
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich



SS 2004 Prof. R. Wattenhofer / Fabian Kuhn / Regina O'Dell

Exam
Principles of Distributed Computing

Monday, October 4th, 2004

Do not open or turn until told so by the supervisor!

Notes

There is a total of 90 points. The number of points is given before each individual question in parentheses. The total for each group of questions is indicated after the title.

Your answers may be in English or in German. Algorithms can be specified in high-level pseudo code or as a verbal description. You do not need to give every last detail, but the main aspects need to be there. Big-O notation is acceptable when giving algorithmic complexities. However, give algorithmic complexities as tight as possible.

Points

Please fill in your name and student ID before the exam starts.

Name	Legi-Nr.

Question Nr.	Achieved Points	Max Points
1		21
2		23
3		12
4		15
5		19
Total		90

1 Publicity Management at the LSS (21 Points)

Big boss W, who has recently become head of the LSS (the organization from Exercise 3), is eager to bring modern management theories to the company. To that end, he needs to gather vital information about his agents. Recall that each member of the LSS can communicate only with his direct superior and his direct subordinates over a secure phone line. W sits on top of this tree hierarchy and initiates the investigations. Assume that agents are distributed all over the world and are not synchronized with each other.

Since international phone calls over secure mobile phone lines are very expensive, agents can only send very short messages each time. Specifically, if there are $n > 1$ members in the entire LSS, each message must be bounded by $O(\log n)$ bits. As usual, we ask for efficient (best possible) algorithms.

a) (3) Having worked many years at a well-known consulting company, W knows that time is money. What is the definition of time complexity of an asynchronous distributed algorithm in general?

b) (4) Speaking of money: W wants to know the average salary in the company. Devise an efficient algorithm, initiated by W, for the LSS structure assuming that each member knows only his or her own salary. What are the time and message complexities of your algorithm?

c) (10) After computing the average salary, W realized that the average was perhaps not the correct measure because only a small number of individuals might earn very much. So he orders another investigation, this time to determine the median of the salaries. (The median of a sequence of ordered numbers $x_1 \leq x_2 \leq \cdots \leq x_n$ is $x_{(n+1)/2}$. You may assume that n is odd.) Under the same conditions as before, propose an algorithm which is as efficient as possible. Give its time and message complexities.

d) (4) After a devastating attack, the entire hierarchy of the LSS has been destroyed. The agents now know some members, not necessarily the same ones as before. W initiates a flooding to rebuild a tree hierarchy. All agents must then report back over the new hierarchy. Assuming a synchronous model, what is the time complexity of this algorithm if we consider the loose collection of agents as a symmetric connected graph G (again, there are still n agents)? What if the communication is asynchronous: What is the time complexity then?

2 Sorting Networks (23 Points)

a) (2) What is the point of studying sorting networks? In other words, why do not all the nodes simply send their input to one central node which can sort quickly (assume local computations to be "free") and then distribute the sorted values back to the nodes?

For each of the following questions, prove or disprove the given claim.

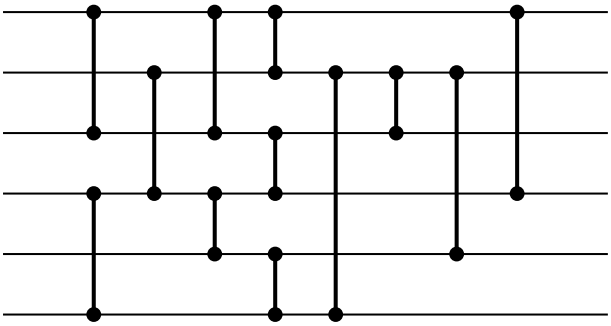


Figure 1: Network for Question 2.

b) (3) The network of 6 wires and 12 comparators in Figure 1 above is a sorting network, that is, it sorts each input sequence of numbers correctly.

c) (2) Given any correct sorting network, adding another comparator at the end destroys the sorting property.

d) (2) Given any correct sorting network, adding another comparator at the front does **not** destroy the sorting property.

e) (4) Every correct sorting network needs to have at least one comparator between each two consecutive wires.

f) (5) A network which contains all $\binom{n}{2}$ comparators between any two of the n wires, in whatever order they are placed, is a correct sorting network.

g) (5) Given any correct sorting network, adding another comparator anywhere does not destroy the sorting property. (Hint: Study examples with a small number of wires.)

3 Broadcast (12 Points)

Recall the notions of consistent broadcast and reliable broadcast in an asynchronous network of n nodes with up to $t < n/3$ Byzantine faults.

The *message complexity* of a protocol is the the number of messages sent by all non-faulty nodes, and the *bit complexity* is the number of bits sent by all non-faulty nodes. (The restriction to non-faulty nodes is needed because the behavior of faulty nodes cannot be controlled.)

Assume a cryptographic digital signature to have length k bits.

a) (3) What are the message and bit complexities of Bracha's implementation of reliable broadcast, when a message m is broadcast?

b) (3) What are the message and bit complexities of the echo broadcast protocol for consistent broadcast?

c) (6) Assume that the nodes use a threshold signature scheme with combination threshold $r = \lceil \frac{n+t+1}{2} \rceil$: This means that every node can produce a share of a digital signature on a particular message, every node can verify that a signature share from another node is correct, and given r correct shares on a message, the digital signature on a message can be assembled.

I) Describe how the echo broadcast protocol can be made more efficient using threshold signatures. What are the resulting message and bit complexities?

II) Can also Bracha's reliable broadcast protocol be made more efficient using threshold signatures? If yes, how?

4 Ivy (15 Points)

a) (2) Give 2 applications for which the Ivy protocol can be used.

Consider the tree for the Ivy protocol in Figure 2 below. The token is held by the circled node labeled r.

b) (1) Draw in the pointers for this initial state.

c) (9) Assume that there are six concurrent requests placed by the nodes v_1 through v_6. Assuming a synchronous execution of Ivy, give the order of serviced requests. Draw the final tree/pointers.

d) (3) Somebody claims to know an infinite sequence of requests for the tree of Figure 2 such that the time to service a request is at least 4 on average. Describe such a sequence of requests or argue why there cannot be such a sequence.

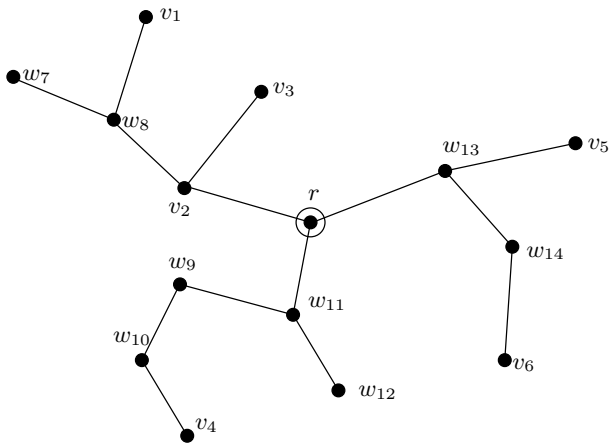


Figure 2: Tree for Question 4.

5 Edge Coloring (19 Points)

A proper edge coloring of a graph is an assignment of colors (numbers) to the edges such that no two neighboring edges have the same color. In this problem, we analyze the following synchronous edge coloring algorithm. We assume that all nodes know Δ, the maximum degree of the network graph G. Further, we assume that all nodes u have unique IDs $\mathrm{id}(u) \in \mathbb{N}$.

Algorithm 1

1. Each node assigns unique values from $\{0, \ldots, \Delta - 1\}$ to all adjacent edges, that is, each edge $e = (u, v)$ is assigned two numbers x_u^e and x_v^e from its two endpoints u and v.
2. Assume that $\mathrm{id}(u) > \mathrm{id}(v)$. The color of an edge $e = (u, v)$ is computed as $\mathrm{color}(e) = \Delta \cdot x_u^e + x_v^e$.

a) (2) How many colors does Algorithm 1 need?

b) (2) What is the time complexity of Algorithm 1?

c) (4) Argue why the time complexity of Algorithm 1 implies that it cannot produce a proper coloring.

d) (5) How many neighboring edges of the same color can an edge have?

e) (6) Give an asymptotically optimal (time complexity) algorithm which properly colors the edges of the network graph with roughly (up to a constant factor) the same number of colors as Algorithm 1. (Hint: Combine Algorithm 1 with an algorithm from the lecture.)