Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

FS 2012

Prof. Dr. Roger Wattenhofer
P. Brandes, K.-T. Förster, S. Holzer,
B. Keller, T. Langner, J. Seidel, J. Uitto,

Exam
Principles of Distributed Computing

Wednesday, August 8, 2012
9:00 – 11:00

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

The exam lasts 120 minutes, and there is a total of 120 points. The maximal number of points for each question is indicated in parentheses. Your answers may be in English or in German. Be sure to always justify your answers. Algorithms can be specified in high-level pseudocode 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.

Please write down your name and Legi number (your student ID) in the following fields.

Name	Legi-Nr.

Points

Question Nr.	Achieved Points	Maximal Points
1		16
2		15
3		31
4		22
5		36
Total		120

1 Multiple Choice (16 Points)

Evaluate each of the following statements in terms of correctness. Indicate whether a statement is true or not by ticking the corresponding box. Each correct answer gets 1 point, **each wrong answer gets -1 point**. An unanswered statement gets 0 points. If the sum of collected points is negative you get 0 points for this question set.

Statement	**true**	**false**
If a graph has a *single* node with degree of $\log n$, then the Diameter is *for sure* in $O(n/\log n)$.	☐	☐
The degree of a n-node pancake graph is in $O(\log \sqrt{n}/\log\log n)$.	☐	☐
The Ramsey number R(4,4) is 8.	☐	☐
If the nodes in a ring know that the number of nodes is even, the ring can be 2-colored in $n/5$ rounds.	☐	☐
At least one color of a valid k-coloring of a graph with chromatic number k (i.e. k colors are optimal) is a maximal independent set.	☐	☐
There exists a sorting algorithm for the $m \times m$ grid that sorts within m rounds.	☐	☐
A bitonic sequence sorter has depth $\log^2 n$.	☐	☐
To glue two open schedules together, both need to have more than one open edge.	☐	☐
Protecting a critical section with the Test-and-Set algorithm fulfills the *unobstructed exit* property.	☐	☐
It is possible for every node in a packet radio network to broadcast successfully once in $\mathcal{O}(\sqrt{n})$ time slots.	☐	☐
The probability to get at least 51% of the times heads if you flip a fair coin n times is at least $\frac{1}{\log n}$	☐	☐
A synchronous distributed algorithm can be executed in an asynchronous environment without changing the asymptotic *time complexity* of the algorithm.	☐	☐
A pseudo-local problem is easier than a strictly local problem.	☐	☐
We learned a MIS algorithm for general graphs that takes $\mathcal{O}(\log n)$ time. It is an open question whether this is optimal.	☐	☐
A MIS algorithm that only takes $\mathcal{O}(\log\log n)$ time contradicts a known lower bound.	☐	☐
The main graph family presented in the reading assignment is regular, that is, each node has the same degree.	☐	☐

2 Birthday Card (15 Points)

Let $G = (V, E)$ be a undirected connected graph with $|V| = n$ and $|E| = m$. Every node has a unique ID, knows its neighbors and their ID, but does not know more about the graph.

A) [6] Assume that one node v has a token that has to be signed by every other node with its ID and given back to v at the end. A real-life example is a birthday-card: one person buys it, it gets handed to all other people (don't cut the card into pieces!) and then it is returned to the buyer of the card. Describe an algorithm that passes the token around. How many messages are sent during the execution of your algorithm?

Remark: Solutions with fewer messages earn more points.

B) [5] Assume that there are non-negative weights on the edges. If the token traverses an edge for the first time in any direction it costs the weight of the edge, but further traversals are free. Describe an algorithm for the problem of **A)** minimizing the cost.

C) [2] Could you actually reduce the amount of messages in **A)** for any input graph by more than a factor of 2 if you could split the messages – a.k.a. cutting the birthday-card and gluing it back together?

D) [2] Can you reduce the cost in **B)** by this at all?

3 Leader Election (31 Points)

In the lecture we saw that leader election can be a rather difficult problem to solve. The goal of this task is to gain some more insights whether leader election is possible.

We assume nodes communicate in synchronous rounds. In each communication round, at most $O(\log n)$ bits may be sent per message to each neighbor in the undirected communication graph $G = (V, E)$.

A node that terminates may no longer send any messages or change its internal state. The whole algorithm terminates, when all nodes have terminated. Once the algorithm terminates, there shall be exactly one node that is in the "leader" state. All other nodes must know that there is a unique leader.

A) Let us first consider some special cases of *anonymous* networks, where nodes do not have unique identifiers. More precisely, all nodes start the algorithm in the same state.

1) [10] Consider the special case of a complete bipartite graph G. This means that G is a graph with $n = |V_1| + |V_2|$ nodes, and the edge set contains exactly the $|V_1| \cdot |V_2|$ edges between those two sets of nodes. More precisely, the set of nodes V is the disjoint union of V_1 and V_2, and the edges are $E = \{\{u, v\} : u \in V_1, v \in V_2\}$.

Describe a uniform, randomized algorithm that solves the problem, and explain why it works. What is the expected running time of your algorithm?

2) [2] Can one also solve the problem on arbitrary bipartite graphs, i.e., where $E \subseteq \{\{u, v\} : u \in V_1, v \in V_2\}$? Explain!

B) In the lecture we saw some special cases of networks with unique node identifiers, in which leader election can be solved deterministically. Our goal is now to find a *fast deterministic* algorithm for arbitrary networks with unique node identifiers.

1) [3] Let D denote the *diameter* of the communication network. Show that leader election takes time at least $\Omega(D)$, or, equivalently, why any algorithm that takes less than some time linear in D cannot accomplish our goal.

2) [16] Give a deterministic, uniform algorithm that elects a leader in $O(D)$ communication rounds. You can assume that a node identifier is small enough to be transmitted in one message, but the nodes may not be numbered from 1 to n. This means it is not sufficient to just use the "agree on node 1 to be the leader"-algorithm.

Explain, why your algorithm is correct, and why it does not take longer than $O(D)$ rounds until all nodes terminate. Also, give an upper bound for the worst case message complexity of your algorithm and prove it.

Recall: An algorithm is said to be uniform, if it can be used for all specified networks without modification. (Nodes do not "know" the size of the network nor any other parameter of the network.)

4 Maximal Independent Set and Coloring (22 Points)

In this short section we are going to look at coloring nodes. A coloring is valid, if and only if it holds for all nodes that any two neighboring nodes have different colors. Nodes have unique IDs and know the IDs of their neighbors.

A) [3] Assume that a maximal independent set (MIS) was constructed on a ring. Every node knows if it is in the MIS or not. Give a uniform algorithm that 3-colors the ring in 1 round.

B) [6] We now expand to other graphs that can be colored with 3 colors. Assume that there is a MIS. Can we still compute a valid 3-coloring in 1 round?

C) [3] Let us turn the question around: Assume you have a valid k-coloring of the nodes (colors from 1 to k), with k being a constant. Can we compute a MIS in $O(1)$ rounds?

Now the situation changes a bit: The input graph is a tree with at least 4 nodes and has a constant maximum degree Δ. The nodes are deaf and blind, i.e., they cannot communicate with each other and cannot see the color of their neighbors. You are allowed to use randomization. With their limited capabilities, the best the nodes can do is to pick a color uniform at random from $\{1, \ldots, 4\}$.

D) [3] How many edges are part of a valid coloring, i.e., for how many edges $(u, v) \in E :\Rightarrow c_v \neq c_u$ holds in expectation.

E) [5] The result from above is not really satisfying. How many colors do you need if you want that at least 95% edges are part of a valid coloring with high probability (in the number of edges, i.e., with probability $1 - m^{-\alpha}$ for constant α)? You can use

$$Pr[X \leq (1-\delta)E[X]] \leq e^{-E[X]\delta^2/2} \qquad \text{for all } 0 < \delta \leq 1$$

in your proof.

F) [2] The algorithm above is an example of a Monte Carlo algorithm, i.e., an algorithm which has a deterministic running time but does not guarantee that its output is correct. There exist also so called Las Vegas algorithms whose output is guaranteed to be correct but the runtime is only achieved with a certain probability. We had several Las Vegas algorithms. Give two examples.

5 Sorting Networks (36 Points)

A) [5] You are given the labelled hypercube H_3 of dimension 3 in Figure 1 and you are asked to construct a correct sorting network with eight wires using comparators only between wires whose corresponding vertices in H_3 are connected by an edge. (For example, you may compare wire 0 and 1 but not wire 1 and 2). Explain why this is not possible.

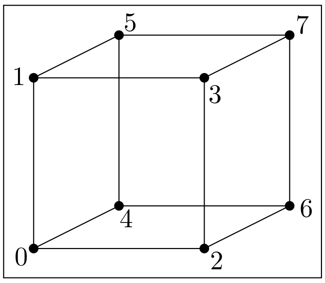

Figure 1: Hypercube H_3 of dimension 3

B) [5] Now you may change the labelling of the vertices in H_3. Prove that this allows you to construct a correct sorting network using comparators only between wires whose corresponding vertices in H_3 are connected.

C) [5] Argue how to assign labels to a hypercube H_d of dimension d to allow the construction of a correct sorting network.

D) [5] Now, instead of normal comparators, you may use *directed* comparators that move the larger element to the upper (↑) or to the lower wire (↓) of the two connected wires. Show how you can construct a correct sorting network with three wires (named 0, 1, and 2), if it is forbidden to have a comparator between wire 1 and 2.

Remark: Solutions with fewer comparators earn more points.

E) [8] Explain how you can transform a sorting network using directed comparators into a sorting network using only normal comparators.

F) [8] Give a correct sorting network with four wires using directed comparators only between wires whose corresponding vertices in H_2 are connected by an edge.

Remark: Solutions with fewer comparators earn more points.

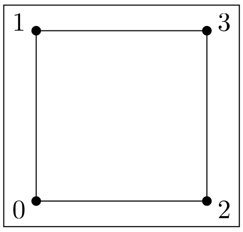

Figure 2: Hypercube H_2 of dimension 2