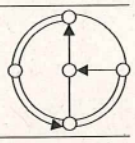
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich



FS 2010

Prof. R. Wattenhofer
C. Lenzen

Exam
Principles of Distributed Computing

24.08.2010

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

Notes

There is a total of 120 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 pseudocode or as a verbal description, unless otherwise mentioned. 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. Unless stated otherwise, you need to prove your claims.

Points

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

Name	Legi-Nr.

Question Nr.	Achieved Points	Max Points
1		36
2		24
3		25
4		35
Total		120

1 Models of Distributed Computation (36 Points)

a) (3) What do we mean by "time complexity of a synchronous distributed message passing algorithm"? How is this notion generalized to asynchronous systems?

b) (3) If the system is fault-free, is there a task for which the fastest synchronous algorithm has a smaller (worst-case) time complexity than the fastest asynchronous algorithm?

c) (3) What is the time complexity of a self-stabilizing algorithm?

d) (13) We say that Model A is as least as powerful as Model B if one can solve all problems that are solvable in B also in A. Order the following deterministic models of distributed computation by their power:
(i) port-numbering,[1]
(ii) anonymous nodes,
(iii) unique identifiers of size $O(\log n)$, where n is the number of nodes in the graph,
(iv) port-numbering + (arbitrarily) oriented edges.[2]

Prove that your order is strict[3] (there are simple examples, with only a few nodes)!

e) (4) Assume now that in the setting of Part **d)** nodes may make use of randomization, i.e., each node has an unlimited source of random bits. Does anything change if we require that time complexity, approximation guarantees, and success of algorithms are certain?

f) (6) Prove that all four models from Part **d)** become equally powerful if we are satisfied with running times, approximation ratios, and success guarantees that hold with probability at least $1 - 1/n$.

g) (4) Suppose you have an excellent algorithm designed for the message passing model, but need to run it on a shared memory system. Is it possible to transfer your algorithm to the new system?

[1] A port-numbering maps for each node its outgoing edges one-to-one to the numbers $1, \ldots, \delta$, where δ is the degree of the node.

[2] Communication is bidirectional, but the edge is "pointing" towards one of the incident nodes.

[3] The order "A is as least as powerful as B" is strict if A and B are not equally powerful, i.e., if "B is as least as powerful as A" does not hold.

2 Greedy Routing in the Augmented Ring (24 Points)

Consider the following network $G = (V, E)$. We have $V := \{1, \ldots, n\}$, i.e., we identify nodes with their identifiers $1, \ldots, n$. The edges are directed and we define that $E := E_0 \cup E_1$, where $E_0 := \{(i, (i+1) \bmod n) \mid i \in \{1, \ldots, n\}\}$. The set E_1 is random and contains for each node exactly one outgoing link. The probability of node $i \in \{1, \ldots, n\}$ connecting to node $j = (i + 2^k) \bmod n$, $k \in \{1, \ldots, \lfloor \log n \rfloor\}$ is $1/\lfloor \log n \rfloor$, independently of the other nodes' outgoing random links.

a) (5) Draw an example of such a network containing 10 nodes. Explain in an informal way how the network is constructed for arbitrary values of n. Aim at an intuitive description and a graphical representation supporting comprehension.

b) (6) Formulate a greedy routing algorithm with expected number of hops of $O(\log^2 n)$ that relies on local knowledge only to decide which of the links of a node to use next. Does your algorithm always terminate?

c) (5) Having bounded the expected number of hops by $O(\log^2 n)$, how can you obtain a bound of $O(\log^2 n)$ on the number of hops that holds with high probability, i.e., probability at least $1 - 1/n^c$ for any predefined constant $c > 0$? (You do not have to give explicit formulas here. Naming the technique and explaining why it is applicable is sufficient.)

d) (4) Having bounded the number of hops by $O(\log^2 n)$ with high probability, infer a bound of $O(\log^2 n)$ on the diameter of the network that also holds with high probability!

e) (4) Does the greedy routing scheme still work if nodes or links may fail permanently?

3 Shared Object with Sequential Writes (25 Points)

Assume the network consists of a rooted tree. The root holds an object. Frequently, nodes want to read the object, whereas write accesses are rare. If a node wants to access the object, but has not obtained a copy yet, it executes the algorithm given below. An inner node does also execute the algorithm whenever it receives a message from a child.

```
1: if not root then
2:    if not requested object before then
3:       send request to parent
4:    end if
5:    wait until copy of the object received from parent
6: end if
7: for all children that requested the object do
8:    send copy of the object to child
9: end for
```

a) (4) For the synchronous network given in Figure 1, how many messages are sent if the nodes labeled $1, 2, 3, 4$ simultaneously request a read? How many messages could at most be sent if those requests happened at arbitrary times and the system was asynchronous?

b) (4) Show that the number of sent messages is at most two times larger than the one of an optimal solution (where an omniscient scheduler knows the whole sequence of requests in advance).

c) (5) If a write occurs, all old copies of the object need to be invalidated. Give an algorithm in pseudocode using the fewest possible number of messages to do so, i.e., after having fetched a copy of the object and modified it, the respective node initiates the invalidation routine. Specify what information the nodes need to store to execute your routine. You may assume that no node accesses the object while the write is in progress.

d) (4) Criticize your solution from part **c)**. What further properties should a well-crafted system provide, and why?

e) (8) Design a system that permits a greater amount of concurrency. We want that during a write operation nodes may concurrently read the object (note that there still is only one write operation in progress at any time). Reads will be executed as before. Define an appropriate consistency guarantee on reads and writes and explain your choice. Give the respective write algorithms. Again, the algorithm shall use as few messages as possible.

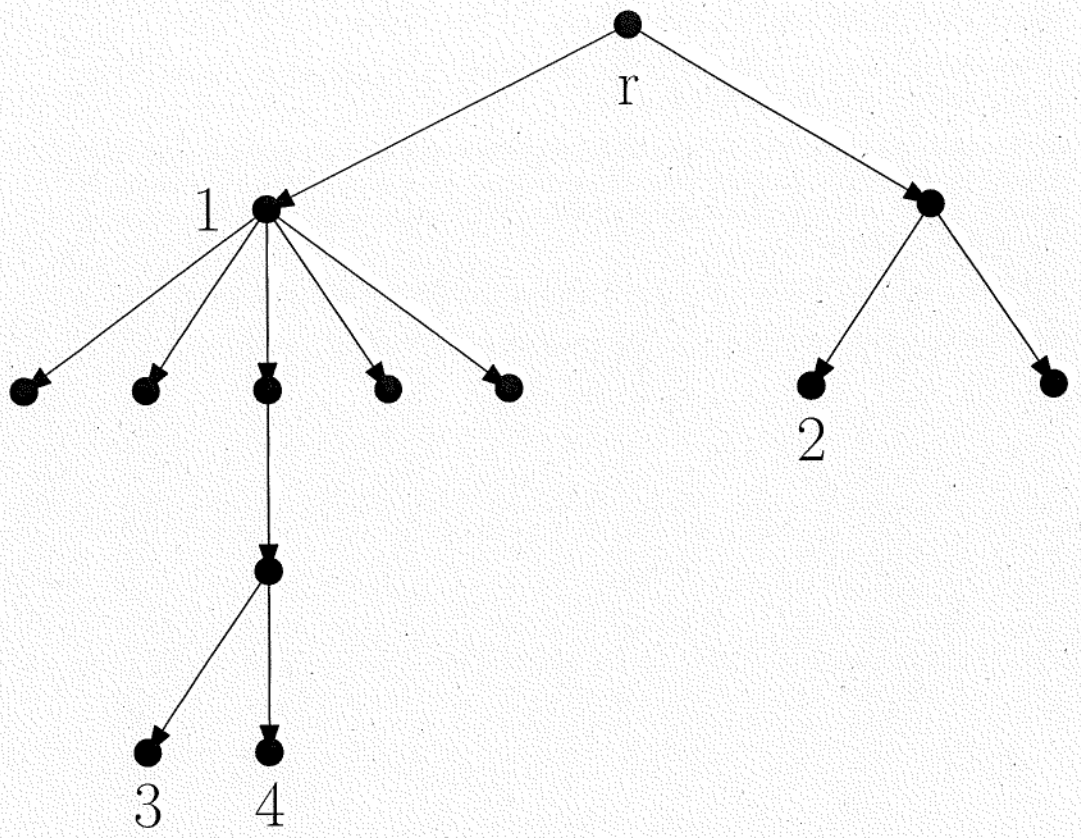


Figure 1: The root of the tree is r. Four read requests are issued at nodes 1 to 4.

4 Coloring and Matching (35 Points)

Consider synchronous deterministic algorithms, where nodes do *not* have unique identifiers. Instead, they use "private" identifiers to refer to their neighbors. Denoting by Δ the maximum degree in the graph, each node refers to each of is neighbors by a different number which is encoded with $O(\log \Delta)$ bits. Note that two neighbors of a node might refer to it by the same value; we just guarantee that a single node assigns different values to its neighbors! In the following, you may assume that nodes know the maximum degree Δ.

a) (9) Suppose each node v is given a color $c(v) \in \{1, \ldots, k\}$, such that no two neighbors have the same color. Show how to color each edge e by a color $e(v) \in O(k\Delta^2)$ such that no node has two edges of the same color. Your algorithm should terminate within $O(1)$ rounds.

b) (4) Assume you colored the edges with $O(k\Delta^2)$ colors. Reduce the number of colors to $2\Delta - 1$. How long does your algorithm take?

c) (4) Prove that a coloring of the edges does not permit a node coloring as given in Part **a)**, for *any* bound on the number of colors!

d) (6) A matching is a subset of the edges such that no two edges have an endpoint in common. A maximal matching is a matching to which no edge can be added without destroying the property to be a matching. Given an edge coloring with k colors, show how to find a maximal matching in $O(k)$ rounds.

e) (6) Given a maximal matching in a graph of maximum degree 2, find an edge coloring.

f) (6) Given a maximal matching, is it always possible to find an edge coloring?