ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

FS 2011

Prof. Dr. Roger Wattenhofer, Dr. T. Locher
R. Eidenbenz, P. Sommer, J. Smula, S. Welten,
J. Schneider, S. Holzer, T. Langner, B. Keller, R. Meier

Exam
Principles of Distributed Computing

Tuesday, August 16, 2011
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, 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.

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		15
2		20
3		18
4		20
5		32
6		15
Total		120

1 Multiple Choice (15 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**
In the LOCAL model, local computation is only accounted for if it takes longer than the average delay in the network.	☐	☐
There is a constant time algorithm to compute a constant approximation of a minimum vertex cover (MVC) for the class of graphs where all nodes have constant degree.	☐	☐
To deterministically color a ring in one round, there are $\Omega(\sqrt{n})$ colors needed.	☐	☐
To deterministically color a ring in 10 rounds, there are $\Omega(\log^{(3)} n)$ colors needed.	☐	☐
It takes $\Omega(\log^* n)$ time to compute a dominating set on a ring deterministically.	☐	☐
The Ramsey number $R(4,4)$ is strictly smaller than 500.	☐	☐
An augmented grid with parameter $\alpha = 0$ has a smaller diameter than an augmented grid with parameter $\alpha = \infty$.	☐	☐
In an augmented grid each node is incident to at most 5 bi-directional links.	☐	☐
Greedy routing from a node u to another node v in an augmented grid of size $m \times m$ takes at most m times more steps than $distance(u, v)$.	☐	☐
There exists a two player rumor game where the player that plays second can always win.	☐	☐
In a d-dimensional hypercube, the number of shortest routes between two peers is upper bounded by $d!$ ("d factorial").	☐	☐
Peers with a constant degree cannot form an overlay with a diameter that is logarithmic in n.	☐	☐
Hypercubes are better than butterfly networks regarding routing fault tolerance.	☐	☐
A set of vertices in a graph is a vertex cover if and only if its complement is an independent set.	☐	☐
If the graph of a dynamic network is 72–interval connected then the graph is connected at any time.	☐	☐

2 Diameter on Trees (20 points)

Develop *asynchronous* message passing algorithms that compute the network diameter on trees for two models with bounded message size. Assume the tree nodes have unique IDs from 1 to n, and–opposed to the normal message passing model–the message size is bounded by $O(\log n)$ bits where n is the number of nodes in the tree.

A) [10] Give an algorithm that computes the diameter as fast as possible under the given model. Show that your algorithm is correct, and analyze its time and message complexity.

B) [10] Now assume the All-to-All model where any node can exchange messages of size $O(\log n)$ with any other node. Give an algorithm that computes the diameter of the tree as fast as possible in this model. Show that your algorithm is correct, and analyze its time and message complexity.

3 Shared Clustering Coefficient (18 points)

Definition

The *local clustering coefficient* $C(v_i)$ of a vertex v_i is defined as the probability that a randomly chosen pair of neighbors of v_i are neighbors of each other, i.e.,

$$C(v_i) = \frac{\#\text{ edges between neighbors of } v_i}{\binom{|\mathcal{N}_{v_i}|}{2}},$$

where $\mathcal{N}_{v_i}$ is the set of neighbors of v_i. The *clustering coefficient* $C(G)$ of a graph $G = (V, E)$ is the average of the local clustering coefficients of all the vertices $v_i \in V$.

For this task we look at the parallel computation of clustering coefficients of a graph $G = (V, E)$ using a set of n processor cores. The graph G has $|V| = n$ nodes and a node degree of at most 10. All processor cores can access a read-only shared memory M. This memory contains the graph G in the following form: For each vertex $v_i \in V$, the memory holds a list M_i containing the indices of v_i's neighbors, i.e. $M_i[j]$ is the index of the j-th neighbor of v_i. Note that the size of a list M_i equals the degree of v_i.

A) [5] Give a shared memory algorithm to compute all local clustering coefficients $C(v_i)$, $i = 1, \ldots, n$, in G (assuming that all processor cores can run in parallel). What is your algorithm's time complexity?

B) [5] Once the local clustering coefficients have been computed, the graph clustering coefficient can be computed by averaging them. Assume that we have one shared read-write memory register R of capacity $O(\log n)$ bits. Given that any processor core $i \in \{1, \ldots, n\}$ knows its local clustering coefficient $C(v_i)$, outline an algorithm that uses R to compute $C(G)$ as fast as possible, and give its time complexity.

C) [5] Assume now that we have an unlimited number of shared read-write memory registers R_k, $k = 1, 2, \ldots$, each with capacity $O(\log n)$ bits. Again, given that any processor core $i \in \{1, \ldots, n\}$ knows $C(v_i)$, outline a fast algorithm that computes $C(G)$ using these registers, and give its time complexity.

D) [3] Is your solution in **C)** asymptotically optimal? Give an informal explanation why it is optimal, or why it is not.

4 Distributed Sorting (20 points)

A) **1)** [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 that uses comparators only between wires whose corresponding vertices in H_3 are connected by an edge. (For example, you may compare wire 0 and 2 but not wire 1 and 2). Explain why this is not possible.

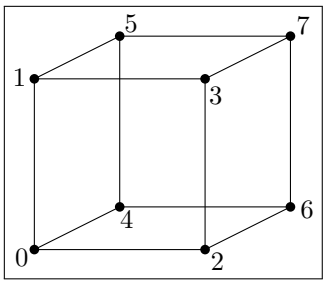

Figure 1: Hypercube H of dimension 3

2) [5] Now you may change the labelling of the vertices in H_3. Prove that this allows you to construct a correct sorting network only using comparators between wires whose corresponding vertices in H_3 are connected.

3) [5] Argue how can to assign labels to a hypercube H_d of dimension d to allow the construction of a correct sorting network.

4) Now, instead of normal comparators, you may use *directed* comparators that move the larger element up or down, depending on their orientation.

□

Figure 2: Hypercube H of dimension 3

B) [10] Let T_n be a complete balanced binary tree with n nodes, $n = 2^h - 1$, $h \in \mathbb{N}$, every node holding exactly one value. The values are to be sorted such that for every node v the following holds: All nodes in its left subtree are smaller and all nodes in its right subtree are larger than v's value. Every node can send/receive $O(1)$ messages per round. Messages can only contain $O(1)$ values. However, nodes may store an unlimited number of values.

- Give a lower bound on the time complexity of any algorithm that sorts the values in T_n.
- Devise a fast algorithm that sorts the values in T_n, show its correctness, and give its time complexity.

C) [10] We still consider complete balanced binary trees as before, but now we want the nodes to satisfy another sorting property: For any node v it must hold that v's value is not larger than the value of either of its children.

- Give a lower bound on the time complexity of any algorithm that sorts the values in T_n according to this sorting property.
- Devise an algorithm that sorts all values correctly in as little time as possible. Show that your algorithm is correct, and give its time complexity.

5 Radius Algorithm (32 points)

Consider the following randomized algorithm, executed by the nodes of an undirected graph $G = (V, E)$ with $|V| = n$ in the synchronous message-passing model. For simplicity, assume that $n = 2^l$ for $l \in \mathbb{N}$. Each node $v \in V$ has a unique ID, where v may refer to both the node and its ID. Further, let $d(u, v)$ denote the distance between two nodes u and v.

Radius Algorithm

1: **Each node** v executes the following code
2: Select a radius $r_v \in \mathbb{N}$ as follows:

$$r_v = \begin{cases} k & \text{with probability } \left(\frac{1}{2}\right)^{k+1}, \text{ for } 0 \leq k < \log n \\ \log n & \text{with probability } \frac{1}{n} \end{cases} .$$

3: Broadcast the pair (v, r_v) to all neighbors within *hop distance* r_v
4: Wait $\log n$ rounds to collect all pairs of the form (u, r_u) (including the own pair (v, r_v))
5: Among all received pairs, select the node with the highest ID as center $C(v)$
6: **if** $d(v,\ C(v)) = r_{C(v)}$ **then**
7: remain uncolored
8: **else if** $d(v,\ C(v)) < r_{C(v)}$ **then**
9: adopt color $C(v)$

Informally, after v has chosen a center $C(v)$, if v is exactly on the border of the set of nodes reached by the chosen center's radius then v remains uncolored. If it is inside it colors itself with the ID of its center.

In the following exercises, we consider the graph G after an execution of the Radius algorithm.

A) [4] Show that if two adjacent nodes are both colored then they must have the same color.

B) [4] Prove or disprove: It is possible for two adjacent nodes u and w to have the same color while v, a common neighbor of u and w, is uncolored.

C) [4] Prove or disprove: Let R_c be the set of all the nodes with color c. For any two nodes $r, s \in R_c$, there exists a path from r to s that consists only of nodes from R_c.

D) [4] Let R_c again be the set of all the nodes with color c. What is the maximum distance between two nodes in R_c?

E) [4] Prove the following statement for nodes u, v with $d(u, v) < \log n$.

$$\Pr[v \text{ is colored} \mid C(v) = u] = \frac{1}{2}$$

F) [4] Use the result from **E)** to show that for a random node v, we have

$$\Pr[v \text{ is colored}] \geq \frac{1}{2}\left(1 - \frac{1}{n}\right)^n \approx \frac{1}{2e} .$$

Now we execute the Radius Algorithm repeatedly on the remaining uncolored nodes. If a node v gets colored with color $C(v)$ in iteration i, we say it joins *cohort* $(i, C(v))$.

G) [2] How many iterations do we need in expectation until all nodes are colored? Justify your claim briefly!

H) [6] The cohorts established by the iterated Radius Algorithm can be used to color the graph G efficiently so that no two neighbors have the same color. Explain what properties of the cohorts could be exploited for this purpose. How would such an algorithm proceed? And how many colors would the resulting coloring have?

6 Reading Assignment (15 points)

A) [5] Draw the cluster tree CT_0 used in the lower bound proof of the MVC problem in the reading assignment, and give a distributed algorithm that solves MVC on the corresponding graph G_0 with message complexity 0.

B) [5] For the graph G_1 corresponding to the cluster tree CT_1, give a fast distributed algorithm that determines for a node in which cluster C_i it is located.

C) [2] What is meant by "locality-preserving reduction"?

D) [3] What is the *girth* of a graph? Why does the lower bound proof of the reading assignment use a construction of a graph G_k with girth at least $2k+1$?

7 Anonymous Coloring (13 points)

anonymous, non-uniform, synchronous

Algorithm 1 Randomized Anonymous Coloring

```
1: pick a color c_v u.a.r. from {1,2,...,k}
2: send c_v to all neighbors
3: receive neighbors' colors
4: if there is a neighbor colored c_v then
5:    go to step 1
```

A) (2 points) How large must k be to guarantee termination?

B) (2 points) Prove that if Algorithm 1 terminates, the produced coloring is valid!

C) (2 points) What is the expected time complexity of Algorithm 1 on a d-regular graph? You may assume that $k = \alpha d$, where α is a constant larger than 1.
Hint: Split the cases where d is constant, and where d is monotonically growing in n.
Hint: $\lim_{n\to\infty} (1 + x/n)^n = e^x$.

Let us look at the following variant of Algorithm 1:

Algorithm 2 Randomized Anonymous Coloring Alternative

```
1: pick a color c_v u.a.r. from {1,2,...,k}
2: repeat
3:    exchange messages to learn neighbors colors
4:    if there is a neighbor colored c_v then
5:       pick a new color c_v u.a.r. from {1,2,...,k}
6: until coloring is valid
```

Note that the termination condition in Line 6 is a global property. Assume for simplicity that there is an oracle telling each node at the end of each round whether the established coloring is valid or not.

A) Explain the difference of the two algorithms with an example.

B) (2 points) How large must k be to guarantee termination?

C) (2 points) What is the expected time complexity of Algorithm 2 on a line/star graph with $k = 2$?

D) (2 points) How could we get rid of the oracle assumption, and turn Algorithm 2 into a truly distributed algorithm that produces a valid coloring w.h.p.? (2–3 sentences)