Questions Academic year 2010-2011
With reference to Kuhn-Wattenhofer material
http://www.dcg.ethz.ch/lectures/fs10/podc/index.html

Ch.1 Vertex coloring

1.In class we considered a simple distributed algorithm which colors an arbitrary graph with c + 1 colors in *n* synchronous rounds was presented (c denotes the largest degree, *n* the number of nodes of the graph).
a) What is the message complexity, i.e., the total number of messages the algorithm sends in the worst case?
b) Does the algorithm also work in an asynchronous environment? If yes, formulate the asynchronous equivalent to the algorithm, if no, describe why.

2. What is the the vertex coloring problem? Motivate its importance in sensor networks. Present a simple algorithm for vertex coloring and discuss its message and time complexity.

4. Define what is a distributed algorithm and discuss the difference between a synchronous and a asynchronous distributed algorithm. Discuss an example where the difference between synchronous and asynchronous is important.

Ch. 2 Leader election

1. Is leader election possible in a synchronous ring in which all but one processors have the same identifier? Either give an algorithm or prove an impossibility result.

2. Consider an anonymous ring where processors start with binary inputs. You can assume that nodes can distinguish between their neighbors, i.e., when a node *v* receives a message, *v* knows which neighbor has sent the message.
a) Prove that there is no uniform synchronous algorithm for computing the AND of the input bits.
b) Present an asynchronous (non-uniform) algorithm for computing the AND; the algorithm should send $O(n^2)$ messages in the worst case.
c) Present a synchronous algorithm for computing the AND; the algorithm should send $O(n)$ messages in the worst case. What is the time complexity of your algorithm?

3. What is leader election? why it is important? Show that deterministic leader election in an anonymous ring is impossible.

4. Show an algorithm for leader election in a synchronous ring and discuss its correctness and complexity.

Ch. 3 Tree algorithms

1. In preparation of a highly dangerous mission, the participating agents of the gargantuan Liechtensteinian secret service (LSS) need to work in pairs of two for safety reasons. All members in the LSS are organized in a tree hierarchy. Communication is only possible via the official channel: an agent has a secure phone line to his direct superior and a secure phone line to each of his direct subordinates. Initially, each agent knows whether or not he is taking part in this mission. The goal is for each agent to find a partner.
a) Devise an algorithm that will match up a participating agent with another participating agent given the constrained communication scenario. A “match” consists of an agent knowing the identity of his partner and the path in the hierarchy connecting them. Assume that there is an even number of participating agents so that each one is guaranteed a partner. Furthermore, observe that1 the phone links connecting two paired-up agents need to remain open at all times. Therefore, you cannot use the same link (i.e., an edge) twice when connecting an

agent with his partner.

2. We consider another day at the office of the LSS as in Exercise 1. After the above mission was successful, the involved agents collected a large number of sensitive documents. Some agents might have a lot of papers and others have none. Now they need to distribute the documents throughout the agency so that each person in the LSS has the same amount of data to process.
a) Assume that there are *n* agents in the LSS and that there is also a total of *n* documents. Devise a way for the agents to distribute their sensitive data: In the end, each agent should have exactly one document. The communication scheme is the same as above.
b) How good is your algorithm with respect to time and number of messages?
b) What are the time and message (i.e., "phone call") complexities of your algorithm?

3. what is broadcast and convergecast? Why they are important? Present and discuss an algorithm for convergecast.

4. present an algorithm for computing a breadth first search tree (BFS tree) and discuss its complexity and correctness.

Ch. 7 Dynamic Networks
7.1 Recall the counting problem in dynamic networks presented in the lecture. Communication is synchronous, message size arbitrary, and each node has an unique identifier. We want all nodes to learn the number of nodes n.
We assume that the dynamic graph G = (V,E) is 2-interval connected, i.e., for any two subsequent rounds r, r + 1, the ("static") graph (V,E(r) \ E(r + 1)) is connected.
Now we drop the assumption from the lecture that all nodes wake up at the same time. Instead, some node u wakes up by itself, while all other nodes start executing the respective algorithm when they receive the first message.
a) Show that it makes no difference if also other nodes may wake up by themselves, i.e., anything you can do if certainly only one node wakes up is still possible.
b) Devise an algorithm that receives an input k and lets u decide whether k _ n or k > n within O(k) rounds.
Hint: Make u wake up all nodes and collect all identifiers assuming that we have less than k nodes. With a little extra time, one will see more than k identifiers if n > k.

7.2 Let us consider 1-interval connected graph. We have seen in class that counting is impossible with asynchronous start and can be done efficiently if all nodes start the protocol at the same time. Present these two results and explainwhy synchrnocity helps.

Ch. 10 Maximal independent sets.
10.1. In the lecture, we discussed a simple maximal independent set (MIS) algorithm in which the decisions of the nodes are based on their identifiers. The time complexity of this algorithm is $O(n)$. We might hope that if the nodes with the largest degrees, i.e., the largest number of neighbors, decide to enter the MIS, the set of undecided nodes reduces the most. In the following algorithm we try to exploit the knowledge of the node degrees:
Assume that each node knows its degree and also the degrees of all its neighbors. If a node has a larger degree than all its undecided neighbors, it joins the MIS and informs its neighbors. Once a node v learns that (at least) one of its neighbors joined the MIS, v decides not to join the MIS.
Naturally, the algorithm does not make any progress if two or more neighboring nodes share the largest degree. As this is a difficult problem, we will assume in the following that this situation does not occur, i.e., if a node *v* has the largest degree, then no neighboring node has the same degree as *v*.1

a) Draw a graph that illustrates that this algorithm has a large time complexity for trees. Give a lower bound on the time complexity for trees consisting of *n* nodes.
b) Discuss the time complexity of the algorithm where at each round an undecided node decides based on "*the number of edges to undecided nodes*" instead of the degree of undecided neighbours. Do you think it is more efficient?

2. What is the maximal independent set problem? Present and discuss the complexity of a simple algorithm that assumes unique identity of nodes.

3. Present and discuss a randomized algorithm for computing a maximal independent set.

Ch. 12

1. What is a synchronizer and why it is important? Present and discuss the complexity of simple synchronizers alpha and beta discussed in class.

2. Present and discuss the complexity of synchronizer gamma discussed in class (including the network partition phase).