Theory of Distributed Systems

Summer Term 2020

Prof. Dr. Martin Hoefer
Marco Schmalhofer

JOHANN WOLFGANG GOETHE
UNIVERSITÄT
FRANKFURT AM MAIN

Institute of Computer Science
Algorithms und Complexity

Exercise 3

Issued: 12.05.2020
Due: 19.05.2020, **14:15h**

Exercise 3.1. *Dijkstra* (6 = 2 + 2 + 2 Points)

Prove or disprove the following bounds:

For every n-vertex, D-diameter graph $G = (V, E)$, there exists an execution of Dijkstra's algorithm requiring

a) $\Omega(D^2)$ time,

b) $\Omega(|E|)$ messages,

c) $\Omega(nD)$ messages.

Exercise 3.2. *Synchronizers* (5 = 2 + 3 Points)

Consider the *gap* of a synchronizer ν, which is the maximum length of a period that some processor v stays in some pulse p. Formally, we define

$$\text{Time}_{gap}(\nu) = \max_{v,p} t(v, p+1) - t(v, p).$$

a) Design a scenario realizing $\text{Time}_{gap}(\alpha) = \Omega(Diam(G))$, in which some processor is forced to wait for time $\Omega(Diam(G))$ until it can increase its pulse number.

b) Prove that this is the worst possible, i.e., $\text{Time}_{gap}(\alpha) = O(Diam(G))$.

Please turn over!

Exercise 3.3. *Smallest k of m* (5 = 2 + 3 Points)

The following policy is proposed for solving the "smallest k of m" problem described in Section 4.3.1 of Peleg's book:

- At any given moment along the execution, every vertex keeps only the set of (at most) k smallest items it knows.
- In each step, each vertex sends to its parent an *arbitrary* element from this set that has not been sent yet.

a) Prove that using this policy, the k smallest elements must still arrive at the root by time $O(k \cdot \mathit{Depth}(T))$.

b) For every integer $m \geq 1$, give an example of a tree T and an initial distribution of m elements, where $k = \mathit{Depth}(T) = \lceil \sqrt{m} \rceil$ and the above process takes $\Omega(m)$ steps.

Exercise 3.4. *Coloring* (6 = 3 + 3 Points)

Consider the generalization of the tree-coloring algorithm from the lecture to arbitrary, bounded degree graphs. The algorithm concatenates for all $w \in \Gamma(v), w \neq v$ the strings which the tree-coloring algorithm would have chosen as new color of node v if neighbor w was v's parent in the tree. This reduction step, starting with a legal K-bit coloring, yields a $\Delta \cdot (\lceil \log K \rceil + 1)$-bit coloring. Assume that the resulting coloring has exactly this length, i.e., each of the entrys that could possibly be shorter is filled with leading zeros.

a) Prove that the coloring obtained from this reduction step is legal.

b) Show that the minimum number of bits that can be achieved by iterating this reduction step is of order $\Theta(\Delta \log \Delta)$, i.e., prove that there are constants $c > c' > 0$ such that for any length more than $c \cdot \Delta \log \Delta$ the process reduces the length and for a length $c' \cdot \Delta \log \Delta$ the process does not reduce the length.

This result is given as Theorem 7.3.8 in Peleg's book. Note that Peleg's estimate of 2Δ instead of $O(\Delta \log \Delta)$ is a bit too optimistic.

The assignments and further information concerning the lecture can be found at
`http://algo.cs.uni-frankfurt.de/lehre/tds/sommer20/tds20.shtml`

E-mail: `mhoefer@cs.uni-frankfurt.de`, `schmalhofer@em.uni-frankfurt.de`