Prof. Dr. Martin Hoefer
Marco Schmalhofer, Giovanna Varricchio

JOHANN WOLFGANG GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Institute of Computer Science
Algorithms und Complexity

Exercise 6

Issued: 30.11.2021
Due: 07.12.2021, **8:15h**

Please submit your solution in PDF format by sending an email to {schmalhofer,varricchio}@em.uni-frankfurt.de. Make sure that your solution reaches us before 8:15 am! Solutions are discussed on Dec 10th, 10:00h - 12:00h (Zoom Meeting-ID: 963 6309 6725, same password as lecture material).
This is the last sheet on Part I.

For all tasks we consider the synchronous CONGEST-model with message size $c \cdot \log_2 n$.

Consider the following variant of the Mailing Problem, the *Both-Zero Mailing Problem*:

Given a graph G with two specified nodes $s \neq r$ as well as bit-vectors $b^{(s)}$ and $b^{(r)}$ of size k for s and r, respectively. Find out whether there is an index where both bit-vectors are 0, i.e., r wants to find out if there is some i with $b_i^{(s)} = b_i^{(r)} = 0$.

Lemma:
For every $m \geq 1$, the Both-Zero Mailing Problem for $k = m^2$ cannot be solved in time $o(m^2/\log m)$ on the hard graph HG_m by a distributed algorithm.

Exercise 6.1. *Weighted Distances* (6 Points)

Use the above lemma to show that in the class of hard graphs finding any approximation to the weighted distance between s and r takes $\Omega(\sqrt{n}/\log n)$ rounds.

Exercise 6.2. *Maximum Weighted Cycles* (6 Points)

A cycle is a path $C = (v_1, v_2, \ldots, v_k, v_1)$, where $v_i \neq v_j$ for $i \neq j$ (walking along the cycle, every node is visited at most once). In a weighted graph $G = (V, E, \omega)$, a maximum weight cycle is a cycle C such that $\omega(C) \geq \omega(C')$ for any cycle C'. In the MaxWeightCycle problem the goal is to compute the value $\omega(C^*)$ of a max-weight cycle C^* in G; in the distributed setting, every node should be aware of the value $\omega(C^*)$.

Use the above lemma to show that solving MaxWeightCycle in the class of hard graphs takes $\Omega(\sqrt{n}/\log n)$ rounds.

Exercise 6.3. *Lower Bound for Randomized APSP* (10 = 2 + 4 + 4 Points)

Given a bit string $x \in \{0,1\}^{n-3}$, the tree $T_n(x)$ is defined in the following way: there is a root r with a left and a right child, c_l and c_r, respectively. Moreover, there is a set of leaves $L = \{1, \ldots, n-3\}$ and, for each $v \in L$, v is a child of c_l, if $x_v = 0$, and is a child of c_r, otherwise.

In the Locating Leaves (LL) problem, we are given a tree $T_n(x)$, and the goal is to inform the root about the location of each leaf node $v \in L$ (left or right subtree). In particular, an output is a vector $s \in \{c_l, c_r\}^{n-3}$, where s_i denotes the parent of leaf i.

Let $\mathcal{X} = \{0,1\}^{n-3}$ be the set of possible bit strings that can be used to generate an input $T_n(x)$. A randomized bit string X is generated using a probability distribution over $\mathcal{X}$.

Let $\mathcal{A}$ be the set of deterministic distributed algorithms solving LL. A randomized algorithm A is a probability distribution over $\mathcal{A}$.

a) Show that any deterministic algorithm needs at least 2^{n-3} different possible outputs to be correct.

b) Show that for every randomized algorithm A and every randomized input tree $T_n(X)$, it holds

$$\min_{a \in \mathcal{A}} \Pr[a \text{ wrong on input } T_n(X)] \;\leq\; \max_{x \in \mathcal{X}} \Pr[A \text{ wrong on input } T_n(x)] \;.$$

Hint: Consider $\Pr[A \text{ wrong on input } T_n(X)]$.

c) Let X be uniformly distributed on $\mathcal{X}$. Show that there are constants $\alpha, \beta > 0$ such that for any deterministic algorithm $a \in \mathcal{A}$ using at most $t \leq \frac{n-3}{4c \log_2 n}$ rounds, it holds

$$\Pr[a \text{ wrong on input } T_n(X)] \geq 1 - \alpha \cdot 2^{-\beta n} \;.$$

Note that the message size is at most $c \cdot \log_2 n$.

From b) and c) it follows that every randomized algorithm for LL on $T_n(x)$ using $o(n/\log n)$ rounds has exponentially small probability of being correct. Notice that the APSP problem is at least as hard as LL.

E-mail: schmalhofer@em.uni-frankfurt.de, varricchio@em.uni-frankfurt.de