ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institute of Theoretical Computer Science
Mohsen Ghaffari, Angelika Steger, Emo Welzl, Peter Widmayer

Algorithms, Probability, and Computing — Exercises KW50 — HS16

The following exercises will be discussed in the exercise class on December 14, 2016. Please hand in your solutions not later than December 13.

In order to be able to work on these exercises, you will probably need to download the lecture notes (LOCAL.pdf) again; they got updated.

Exercise 1: Reducing the Number of Colors in a Single Round

In Lemma 8.5, we saw a single-round algorithm for reducing the number of colors exponentially. Here, we discuss another such method, which transforms any k-coloring of any rooted-tree to a $2 \log k$-coloring, so long as $k \geq C_0$ for a constant C_0.

The method works as follows. Let each node u send its color $\phi_{old}(u)$ to its children. Now, each node v computes its new color $\phi_{new}(v)$ as follows: Consider the binary representation of $\phi_{old}(v)$ and $\phi_{old}(u)$, where u is the parent of v. Notice that each of these is a $\log_2 k$-bit value. Let i_v be the smallest index i such that the binary representations of $\phi_{old}(v)$ and $\phi_{old}(u)$ differ in the i^{th} bit. Let b_v be the i_v^{th} bit of $\phi_{old}(v)$. Define $\phi_{new}(v) = (i_v, b_v)$. Prove that $\phi_{old}(v)$ is well-defined, and that it is a proper $(2 \log k)$-coloring.

Exercise 2: Coloring Trees That Are Not Rooted

In Theorem 8.4, we saw that on a tree graph where each node knows its parent, a 3-coloring can be computed in $O(\log^* n)$ rounds. This result heavily relies on each node knowing its parent, and no such result is true in unrooted trees, that is, when nodes do not know which neighbor is their parent.

It is known that there exists Δ-regular graphs with girth — that is, the length of the shortest cycle — being at least $\Omega(\log_\Delta n)$ and chromatic number at least $\Omega(\Delta/\log \Delta)$. (This result is due to Bollobás.)

Use this existence to prove that any deterministic $o(\log_d n)$-round algorithm for coloring trees (not rooted) requires at least $\Omega(\Delta/\log \Delta)$ colors.

Exercise 3: Randomized Network Decomposition

Prove Lemma 8.28 (in the lecture we chose $\varepsilon = \frac{1}{2}$):

With high probability, the maximum cluster diameter is at most $O(\log n)$. *Hence, this clustering can be computed in* $O(\log n)$ *rounds, with high probability.*

Exercise 4: More Algorithms for Color Reduction

(a) Design a single-round algorithm that transforms any given k-coloring of a graph with maximum degree Δ into a k'-coloring for $k' = k - \lceil \frac{k}{2(\Delta+1)} \rceil$, assuming $k' \geq \Delta+1$.

(b) Use repetitions of this single-round algorithm, in combination with the $O(\log^* n)$-round $O(\Delta^2)$-coloring of Theorem 8.12, to obtain an $O(\Delta \log \Delta + \log^* n)$-round $(\Delta + 1)$-coloring algorithm.