SS 2007 Prof. R. Wattenhofer / Prof. P. Widmayer / Prof. S. Suri / T. Locher / Y. A. Oswald

Principles of Distributed Computing
Exercise 5

1 Edge Coloring

A proper edge coloring of a graph is an assignment of colors (numbers) to the edges such that no two neighboring edges have the same color. In this problem, we analyze the following synchronous edge coloring algorithm. We assume that all nodes know Δ, the maximum degree of the network graph G. Further, we assume that all nodes u have unique IDs $id(u) \in \mathbb{N}$.

Algorithm 1

1. Each node assigns unique values from $\{0, \ldots, \Delta - 1\}$ to all adjacent edges, that is, each edge $e = (u, v)$ is assigned two numbers x_u^e and x_v^e from its two endpoints u and v.
2. Assume that $id(u) > id(v)$. The color of an edge $e = (u, v)$ is computed as $color(e) = \Delta \cdot x_u^e + x_v^e$.

a) How many colors does Algorithm 1 need?

b) What is the time complexity of Algorithm 1?

c) Argue why the time complexity of Algorithm 1 implies that it cannot produce a proper coloring.

d) How many neighboring edges of the same color can an edge have?

e) Give an asymptotically optimal (time complexity) algorithm which properly colors the edges of the network graph with roughly (up to a constant factor) the same number of colors as Algorithm 1. (Hint: Combine Algorithm 1 with an algorithm from the lecture.)