ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

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

Principles of Distributed Computing
Exercise 5: Sample Solution

1 Edge Coloring

a) Both nodes u and v might choose the maximum color $\Delta - 1$ for edge e. The resulting color of edge e would then be $color(e) = \Delta(\Delta - 1) + \Delta - 1 = \Delta^2 - 1$. Thus, the algorithms needs at most Δ^2 colors (including color 0).

b) The time complexity of Algorithm 1 is constant, as it always terminates after step 2.

c) This algorithm colors the edges of a *ring* in constant time using 3 colors. In order to color the nodes of the ring, we first color the edges and then, assuming that nodes have a sense of direction, let each node adopt the color of its left (or right) edge.[1] This is a contradiction to the $\Omega(\log^* n)$ time lower bound for coloring rings (even if nodes have a sense of direction). Hence, the algorithm cannot produce a proper coloring.

d) Each node v can have 2 adjacent edges of any particular color. For example, if $id(u) < id(v) < id(w)$, let the nodes u, v, w choose the colors $x_u^{(u,v)} := \alpha, x_v^{(u,v)} := \beta, x_v^{(v,w)} := \alpha$, and $x_w^{(v,w)} := \beta$. Both edges (u, v) and (v, w) will get the color $\beta \cdot \Delta + \alpha$. It is easy to see that not more than two outgoing edges of any node can obtain the same color, and thus every edge can only have one neighboring edge of the same color at each endpoint.

e) Because each edge has only one neighboring edge of the same color at each endpoint, the colored graph consists of *lines* of edges colored the same. For each color α, we define two new colors α' and α'' and color all the lines in the graph using its old color and the two corresponding new colors *simultaneously*. As a line can be colored using 3 colors in $O(\log^* n)$ time and all lines are recolored in parallel, we get an algorithm that colors all edges of the network graph G in $O(\log^* n)$ time using a total of at most $3\Delta^2$ colors.

[1] Note that, in a single step, the number of colors can be reduced from $\Delta^2 = 4$ to 3 by simply recoloring all nodes whose color is 4 with any color not used by their neighbors.