Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

FS 2010 Prof. R. Wattenhofer / C. Lenzen

Principles of Distributed Computing
Exercise 12

1 Coloring Rings

In Chapter 1, we proved that a ring can be colored with 3 colors in $\log^* n + O(1)$ rounds. Clearly, a ring can only be (legally) colored with 2 colors if the number of nodes is even.

a) Prove that, even if the nodes in a directed ring know that the number of nodes is even, coloring the ring with 2 colors requires $\Omega(n)$ rounds![1]

Since coloring a ring with 2 colors apparently takes a long time, we again resort to the problem of coloring rings using 3 colors.

b) Assume that a *maximal independent set* (MIS) has already been constructed on the ring, i.e., each node knows whether it is in the independent set or not. Give an algorithm to color the ring with 3 colors in this scenario! What is the time complexity of your algorithm? Deduce from this a lower bound for computing a MIS!

We now want to close the gap between the lower bound of $(\log^* n)/2 - 1$ and the upper bound of $\log^* n + O(1)$:

c) Give an algorithm, based on Algorithm Six-2-Three from Chapter 1, that colors a *directed* ring using 3 colors in $(\log^* n)/2 + O(1)$ rounds![2]

[1] As in the lecture, the message size and local computations are unbounded and all nodes have unique identifiers from 1 to n.

[2] Use the information received from *both* neigbhors to perform 2 rounds of algorithm in each round!