Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

FS 2024
Prof. R. Wattenhofer

Principles of Distributed Computing
Exercise 5

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 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 deterministically with 3 colors in this scenario! What is the time complexity of your algorithm? Deduce from this a lower bound for computing a MIS!

2 Ramsey theory

In the classic example for Ramsey theory ($R(3, 3)$), it is asked how many people you can invite to a party so that are no three people who mutually know each other, and no three people who are mutual strangers. This could be important for planning a party, since three people who mutually do (not) know each other will form their own subgroup during the party and rarely interact with the other guests. What is the largest number of people you can invite?

In more mathematical terms, this is a coloring problem: consider the complete graph K_n with n nodes. Can you assign each edge to one of two colors (for example red and blue), so that there is no K_3 as a subgraph where all edges have the same color? What is the maximum number n for which you can find such a coloring?

a) Show that such an edge-coloring exists K_5, but not for K_6: $R(3, 3) = 6$. This means that you can only invite 5 people to your party.

Parties with only 5 people are, however, pretty boring. You change your constraints to allow bigger parties as follows: in any group of three people, there is at least one mutual stranger to the other two.

b) What is the maximum number of people n for which this constraint can be satisfied?

c) What if we look for a setup of n people that also maximizes the number of pairs that mutually know each other? What would be the maximum number of such pairs that we can obtain?

To avoid raising the number of people that are mutual strangers, we add one more constraint: there should be no group of p people that are mutual strangers.

d) What is the largest number of people (depending on p) that can attend the party under the new constraints?

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