FDS Exercise 10

1 Vertex Coloring

In the lecture, a simple distributed algorithm ("Reduce") which colors an arbitrary graph with $\Delta + 1$ colors in n synchronous rounds was presented (Δ denotes the largest degree, n the number of nodes of the graph).

1. What is the message complexity, i.e., the total number of messages the algorithm sends in the worst case?

2. Does the algorithm also work in an asynchronous environment? If yes, formulate the asynchronous equivalent to the algorithm, if no, describe why.

2 Deterministic Maximal Independent Set

In the lecture, we discussed a simple maximal independent set (MIS) algorithm in which the decisions of the nodes are based on their identifiers. The time complexity of this algorithm is $O(n)$.

We might hope that if the nodes with the largest degrees, i.e., the largest number of neighbors, decide to enter the MIS, the set of undecided nodes reduces the most. In the following algorithm we try to exploit the knowledge of the node degrees:
Assume that each node knows its degree and also the degrees of all its neighbors. If a node has a larger degree than all its undecided neighbors, it joins the MIS and informs its neighbors. Once a node v learns that (at least) one of its neighbors joined the MIS, v decides not to join the MIS.

Naturally, the algorithm does not make any progress if two or more neighboring nodes share the largest degree. As this is a difficult problem, we will assume in the following that this situation does not occur, i.e., if a node v has the largest degree, then no neighboring node has the same degree as v.[1]

1. Draw a graph that illustrates that this algorithm has a large time complexity for trees! Give a (non-trivial) lower bound on the (worst-case) time complexity for trees consisting of n nodes!

2. Construct a graph that shows that the time complexity of this algorithm is even worse for arbitrary graphs than for trees! What is the time complexity?

3 Coloring Rings

We have proved in the Lecture 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. 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![2]

[1] The motivation for this constraint is that if we prove that the time complexity is large even if there is no conflict in each step, then being able to break ties clearly does not help.

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