Network Algorithms: Exercise 5
Coloring + Tree Algorithms

Dr. Stefan Schmid, Arne Ludwig, Srivatsan Ravi

1 Vertex Coloring (10 + 20 + 15 = 45 points)

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).

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

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

c) Assume the graph forms a tree. Argue why the algorithm needs more than $O(1)$ many colors in the worst case.

2 Coloring Rings and Trees (20 points)

Algorithm 15 (Six-2-Three) in the lecture notes colors any (directed) tree consisting of n nodes with 3 colors in $O(\log^* n)$ rounds. Show how the log-star coloring algorithm for trees can be adapted for rings given that the nodes know n.

3 MST Construction in a Clique (15 + 15 + 30 = 60 points)

Recall the MST Definition 3.8 from the lecture notes: The MST is the spanning tree of minimal costs. Also the BFS can be defined for weighted graphs: a BFS is the shortest path tree from a given source. In the lecture you saw how the MST can be computed in a distributed manner (GHS algorithm). If the underlying graph is not a general graph, but a special graph, then faster solutions exist. In the following, you will study the clique network (completely connected) in more detail.

a) Show that there are examples where the MST is different from any BFS tree (for any possible source). Hint: Use the clique with 5 nodes and assign link costs.

b) Give an algorithm which generates a MST in $O(1)$ time complexity and $O(m)$ message complexity (m the number of links) in a clique.

c) Does your algorithm still work if in each round, at most a constant number of link weights and node IDs can be transmitted over any given link? If not, can you come up with a solution which runs in time $O(\log n)$? The message complexety should remain $O(m \log n)$.