Principles of Distributed Computing	03/11, 2020
Exercise 4	
Lecturer: Mohsen Ghaffari	

1 Problem 1, Lower Bound for Locally-Minimal Coloring

For a graph $G = (V, E)$, a coloring $\phi : V \to \{1, 2, ..., Q\}$ is called *locally-minimal* if it is a proper coloring, meaning that no two adjacent vertices v and u have $\phi(v) = \phi(u)$, and moreover, for each node v colored with color $q = \phi(v) \in \{1, 2, ..., Q\}$, all colors 1 to $q - 1$ are used in the neighborhood of v. That is, for each $i \in \{1, \ldots, q - 1\}$, there exists a neighbor u of v such that $\phi(u) = i$.

Exercise

(1a) In the 3^{rd} lecture, we saw a $O(\Delta \log \Delta + \log^* n)$-round algorithm for computing a $(\Delta + 1)$-vertex-coloring in any n-node graph with maximum degree Δ. Use this algorithm to compute a *locally-minimal coloring* in $O(\Delta \log \Delta + \log^* n)$ rounds, in any n-node graph with maximum degree Δ.

Compute a $(\Delta+1)$-coloring, which will be used as a schedule-color. Process the colors of the schedule-color one by one, in $\Delta + 1$ iterations, each time picking a locally-minimal color for all nodes with schedule-color $i \in \{1, 2, \ldots, \Delta + 1\}$.

In the remainder of this exercise, we prove a lower bound of $\Omega(\log n/ \log \log n)$ on the round complexity of computing a *locally-minimal coloring*, for some graphs. We note that these graphs have maximum degree $\Delta = \Omega(\log n)$ and hence, this lower bound poses no contradiction with (1a).

For the lower bound, we will use a classic graph-theoretic result of Erdős [?]. Recall that the girth of a graph is the length of its shortest cycle, and the chromatic number of a graph is the smallest number of colors required in any proper coloring of the graph.

Theorem 1 (Erdős [?]) *For any sufficiently large n, there exists an n-node graph G_n^* with girth $g(G_n^*) \geq \frac{\log n}{4 \log \log n}$ and chromatic number $\chi(G_n^*) \geq \frac{\log n}{4 \log \log n}$.*

Exercise

(1b) Prove that in any locally-minimal coloring $\phi : V \to \{1, 2, ..., Q\}$ of a tree $T = (V, E)$ with diameter d — i.e., where the distance between any two nodes is at most d — no node v can receive a color $\phi(v) > d + 1$.

Suppose for the sake of contradiction that a node v_0 receives a color $k \geq d + 2$. Then, v_0 must have a neighbor v_1 who has color $k-1$. Similarly, v_1 must have a neighbor v_2 who has color $k-2$. Continuing this process, we create a simple path v_0, v_1, v_2, ... of length $k - 1$ whose vertices have colors k, $k - 1$, $k - 2$, ..., respectively. In a tree, any simple path between two vertices is a shortest path between them. Hence, the tree has two nodes at distance $k - 1 \geq d + 1$, which is a contradiction.

(1c) Suppose towards contradiction that there exists a deterministic algorithm $\mathcal{A}$ that computes a locally-minimal coloring of any n-node graph in at most $\frac{\log n}{8 \log \log n} - 1$ rounds. Prove that when we run $\mathcal{A}$ on the graph G_n^*, it produces a (locally-minimal) coloring with at most $Q = \frac{\log n}{4 \log \log n} - 1$ colors. For this, you should use part (1b) and the fact that G_n^* has girth $g(G_n^*) \geq \frac{\log n}{4 \log \log n}$.

Consider running $\mathcal{A}$ on G_n^*. We claim that no node $v \in G_n^*$ can receive a color $k \geq \frac{\log n}{4 \log \log n}$. The reason is as follows. Now imagine running $\mathcal{A}$ on the subgraph G_v of G_n^* induced by nodes within distance $\frac{\log n}{8 \log \log n} - 1$ of v. The algorithm $\mathcal{A}$ must assign the same color k to v, as when $\mathcal{A}$ is run on G_n^* (why?). However, G_v is a tree with diameter at most $\frac{\log n}{4 \log \log n} - 2$ (why?). Hence, by the property proven in (1b), in any valid locally-minimal coloring, the highest color that node v can receive is at most $Q = \frac{\log n}{4 \log \log n} - 2 + 1$.

(1d) Conclude that any locally-minimal coloring algorithm needs at least $\frac{\log n}{8 \log\log n}$ rounds on some n-node graph.

By (1c), if $\mathcal{A}$ always runs in at most $\frac{\log n}{8 \log\log n} - 1$ rounds, it produces a coloring of G_n^* where each node is colored with a color in $1, 2, \ldots, Q$ for $Q = \frac{\log n}{4 \log\log n} - 1$. This is in contradiction with G_n^* having chromatic number $\chi(G_n^*) \geq \frac{\log n}{4 \log\log n}$. Having arrived at the conclusion by assuming that $\mathcal{A}$ always runs in at most $\frac{\log n}{8 \log\log n} - 1$ rounds in any n-node graph, we conclude that algorithm $\mathcal{A}$ needs at least $\frac{\log n}{8 \log\log n}$ rounds on some n-node graph.