CS:5620 Homework 1 Solution, Fall 2016

1(b) We prove the correctness of the algorithm by induction. Let $v_1, v_2, \ldots, v_n$ be a sequence of the nodes in the graph G such that $\mathtt{ID}_{v_1} \geq \mathtt{ID}_{v_2} \geq \cdots \geq \mathtt{ID}_{v_n}$. For any non-negative integer j, let $S_j = \{v_1, v_2, \ldots, v_j\}$. (So $S_0 = \emptyset$.) The inductive claim we make is the following:

> For any non-negative integer j, after j rounds of the algorithm, two properties are satisfied: (i) for all nodes $v \in S_j$, $c(v) \neq \perp$ and (ii) for any edge $\{u, v\} \in E$, if $c(u) \neq \perp$ and $c(v) \neq \perp$ then $c(u) \neq c(v)$.

(Property (i) is saying that after j rounds at least the first j nodes are colored and property (ii) is saying that the partial coloring we get after j rounds is proper.) Note that if the above claim holds for $j = n$, then the algorithm is correct.

The base case of the inductive proof is when $j = 0$ and this is trivial. For the inductive step, suppose that the claim is true for some j. Now consider round $j + 1$. We will show two things about what happens in round $j + 1$: (a) if v_{j+1} is not already colored, it will be colored in round $j + 1$ and (b) all the nodes that are colored in round $j + 1$ will have colors distinct from their neighbors.

(a) If v_{j+1} is not colored before round $j + 1$, then by the inductive hypothesis part (i), all its neighbors with strictly higher IDs have been colored in previous rounds. Since v_{j+1} has no neighbors with the same ID as it, v_{j+1} will color itself in round $j + 1$.

(b) Now consider nodes that are colored in round $j + 1$. Note that no two nodes that are colored in round $j + 1$ can be neighbors. (If they are, then one has a strictly higher ID than the other and the node with the lower ID will wait for a later round to color itself.) Thus any node that colors itself in round $j + 1$, uses a color that is distinct from all neighbors that have been already colored.

Showing (a) and (b) shows that at the end of round $j + 1$, all nodes in S_{j+1} are colored (property (i)) and that the partial coloring, thus far produced, is proper (property (ii)).

(3) We will describe a distributed algorithm in the LOCAL model that establishes the following more general claim:

> In the LOCAL model, we can compute a proper vertex coloring of G using $\chi(G)$ colors in diameter$(G) + 2$ rounds.

We prove this claim, by describing an algorithm.

In Round 1 of the algorithm, each node sends its ID to all neighbors. Using this received information, each node v makes a list $Nbd(v)$ that starts with $\mathtt{ID}_v$ and is followed (in no particuler order) by the IDs of the neighbors of v. $Nbd(v)$ represents the neighborhood of node v. Each node v also initializes three local variables, (i) $Seen(v)$ to $\{\mathtt{ID}_v\}$, (ii) $ToSend(v)$ to $\{Nbd(v)\}$, and $Graph(v)$ to $\{Nbd(v)\}$. The variable $Seen(v)$ keeps tracks of IDs of nodes whose neighborhoods are known to v. The variable $ToSend(v)$ keeps tracks of neighborhoods that need to be sent out in the next round. The variable $Graph(v)$ keeps tracks of input graph, represented as a set of neighborhoods, that node v knows so far.

In each Round j, $j > 1$, each node v does the following.

1. If $ToSend(v)$ is non-empty then, node v sends the set $ToSend(v)$ to all neighbors.
2. Receives messages (that are non-empty sets) from neighbors, though not necessarily from all neighbors.
3. Computes the union of the sets received from neighbors and deletes from this union all $Nbd(u)$ such that $\mathtt{ID}_u \in Seen(v)$. The resulting set is assigned to $ToSend(v)$, it is added to $Graph(v)$, and for every node u such that $Nbd(u) \in ToSend(v)$, ID_u is added to $Seen(v)$.

It is not too hard to show that after diameter$(G)+2$ rounds, no further messages are sent and $Graph(v)$ equals the input graph for all nodes v. Then, each node v, can use the infinite computational power it has at its disposal to compute a proper vertex coloring of $G = Graph(v)$, using $\chi(G)$ colors. At this point each node v knows its color in this optimal coloring[1].

---

[1] There is the technical issue that there could be several optimal colorings and these need not be consistent. We can assume that each node generates colorings in some order (that is the same for all nodes) and nodes use the first optimal coloring generated in this order.