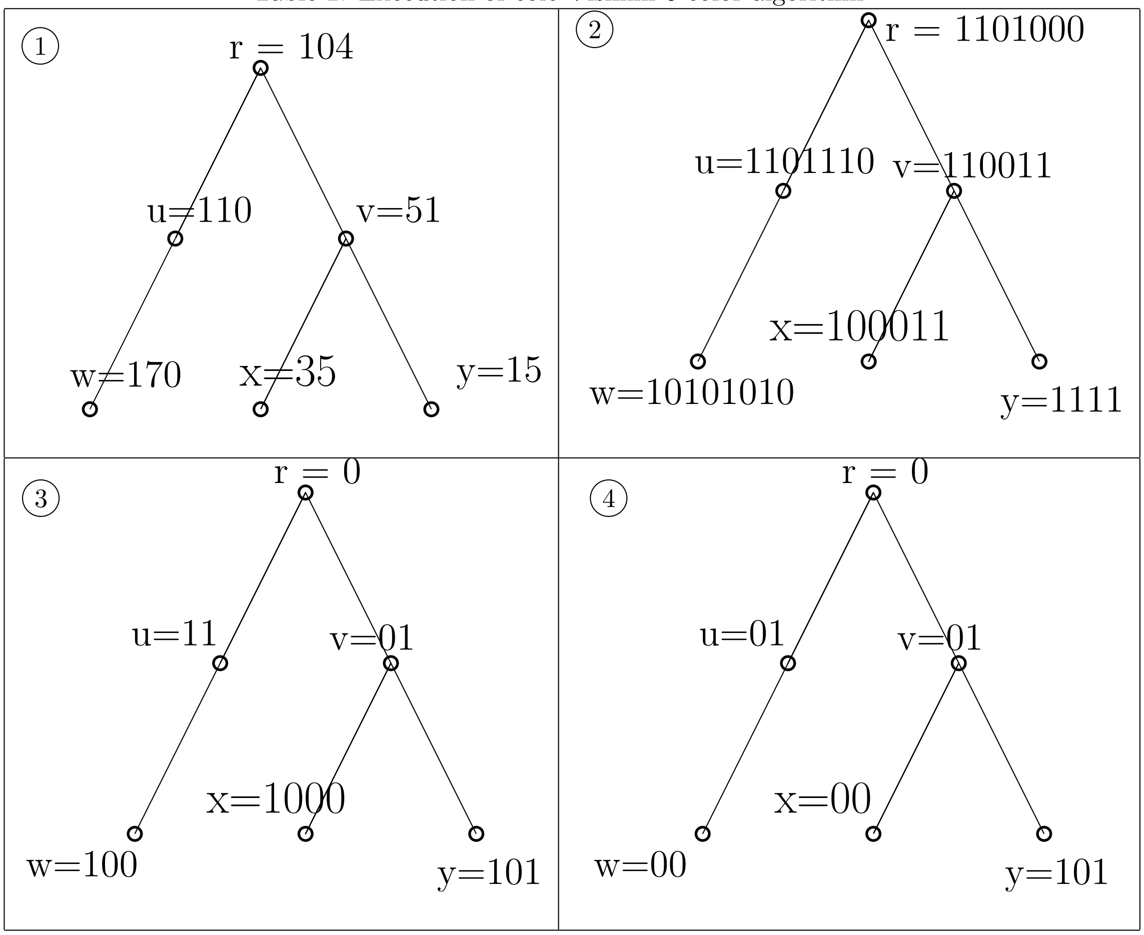
CS:5620 Homework 2 Solution, Fall 2016

Table 1: Execution of cole-Vishkin 6 color algorithm



(1)

(2.a)

$$
\begin{aligned}
Pr[C_u] =& \frac{1}{|P(u)|} \sum_{c \in P(u)} Pr(\overline{W}_{c,N(u)}) \\
Pr[C_u] =& \frac{1}{6} \cdot \left[\left(\frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} \cdot 1 \cdot 1 \right) + \left(1 \cdot 1 \cdot \frac{1}{2} \cdot 1 \cdot \frac{1}{2} \cdot 1 \right) + \left(1 \cdot 1 \cdot 1 \cdot \frac{1}{2} \cdot 1 \cdot 1 \right) + \right. \\
& \left. \left(\frac{1}{2} \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 \right) + (1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1) + (1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1) \right] \\
Pr[C_u] =& \frac{1}{6} \cdot \left(\frac{1}{16} + \frac{1}{4} + \frac{1}{2} + \frac{1}{2} + 1 + 1 \right) \\
Pr[C_u] =& \frac{53}{96}
\end{aligned}
$$

(2.b) $c' = 1$, $c'' = \{3, 4, 5, 6\}$

(2.c) Replacing 1 with 6 in the palette of v_1

$$\begin{aligned}
Pr[C_u] =& \frac{1}{|P(u)|} \sum_{c \in P(u)} Pr(\overline{W}_{c,N(u)}) \\
Pr[C_u] =& \frac{1}{6} \cdot \left[\left(1 \cdot \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} \cdot 1 \cdot 1\right) + \left(1 \cdot 1 \cdot \frac{1}{2} \cdot 1 \cdot \frac{1}{2} \cdot 1\right) + \left(1 \cdot 1 \cdot 1 \cdot \frac{1}{2} \cdot 1 \cdot 1\right) + \right. \\
& \left. \left(\frac{1}{2} \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1\right) + (1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1) + \left(\frac{1}{2} \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1\right) \right] \\
Pr[C_u] =& \frac{1}{6} \cdot \left(\frac{1}{8} + \frac{1}{4} + \frac{1}{2} + \frac{1}{2} + \frac{1}{2} + 1 \right) \\
Pr[C_u] =& \frac{23}{48} = \frac{46}{96} < \frac{53}{96}
\end{aligned}$$

The probability goes down as expected.

(3) In the first phase, we run the Cole-Vishkin algorithm to obtain a $2^{2\Delta}$-coloring in $O(\log^* n)$ rounds. This algorithm runs in the CONGEST model. Now we start the second phase of the algorithm. For each color $c \in \{1, 2, \ldots, 2^{2\Delta}\}$ considered in this order, we process all vertices of color c in parallel. If a vertex v of color c has a neighbor of color $c' \in \{1, 2, \ldots, c-1\}$ in the MIS, then v chooses not to join the MIS; otherwise v joins the MIS. Thus all vertices of color c are processed in $O(1)$ rounds and therefore, Phase 2 runs in $O(2^{2\Delta})$ rounds. Since it is given that $\Delta \leq 10$, $2^{2\Delta}$ is bounded above by a constant (though a somewhat large constant). Therefore, Phase 2 runs in $O(1)$ rounds, also in the CONGEST model and the entire algorithm runs in $O(\log^* n)$ rounds in the CONGEST.
Remark: I did not prove the correctness of the algorithm here, but hopefully it is easy to see.

(4) Node v intitalizes its color $c(v)$ to $\perp$ and a boolean flag $done(v)$ to False. To initialize its palette $P(v)$, node v executes the following 2-round algorithm. The purpose of this algorithm is simply to count the number of nodes in the 2-neighborhood of eacn node v.

1. v **sends** $\texttt{ID}_v$ to all neighbors
2. v **receives** IDs from neighbors; let $N_{ID}(v)$ denote the set of IDs received
3. v **sends** $N_{ID}(v)$ to all neighbors. (In this step messages can be quite large.)
5. v **receives** sets of IDs from neighbors.
6. **for** each neighbor u **do**
7. $\quad N_{ID}(v) \leftarrow N_{ID}(v) \cup N_{ID}(u)$
8. $P(v) \leftarrow \{1, 2, \ldots, |N_{ID}(v)|\}$

After $P(v)$ has been initialized, node v repeatedly executes the following 4-round algorithm.

// Pick a tentative color, if not already permanently colored
1. **if not** $done(v)$ **then**
2. $\quad c(v) \leftarrow$ a color picked uniformly at random from palette $P(v)$
3. $\quad v$ **sends** $c(v)$ to all neighbors

// Even permanently colored nodes should continue to pass on received colors to neighbors
4. v **receives** colors from neighbors; let $N_C(v)$ denote the set of colors received
5. v **sends** $N_C(v)$ to all neighbors. (In this step messages can be quite large.)

// Determine if my color collides with the color of any node in my 2-nbd
6. **if not** $done(v)$ **then**
7. $\quad v$ **receives** sets of colors from neighbors

for each neighbor u that v receives a set of colors from **do**
    $N_C(v) \leftarrow N_C(v) \cup N_C(u)$
// If there is no collision then I become permanently colored
if $c(v) \notin N_C(v)$ **then**
    $done(v) \leftarrow True$
    v **sends** $c(v)$ to all neighbors

//Permanently assigned colors need to be deleted from palletes in 2-nbd of still-active nodes
v **receives** colors from neighbors; let $N_P(v)$ denote the set of colors received
v **sends** $N_P(v)$ to all neighbors. (In this step messages can be quite large.)
if not $done(v)$ **then**
  v **receives** sets of colors from neighbors
  for each neighbor u that v receives a set of colors from **do**
    $P(v) \leftarrow P(v) \setminus P_N(u)$

(5.a) Suppose that the **while**-loop in the above algorithm runs for t iterations, for some non-negative integer t. For $1 \leq i \leq t$, let S_i denote the set S in the ith iteration of the **while**-loop. Thus S_1 denotes the set of nodes in G with degree at most 2. More generally, S_i is the set of nodes that have degree at most 2 in the subgraph of G induced by $S_i \cup S_{i+1} \cup \cdots \cup S_t$.

We now prove by contradiction that $|S_1| \geq n/3$. Otherwise, if $|S_1| < n/3$, then G contains more than $2n/3$ nodes with degree 3 or more. Therefore, the total degree of nodes in $V \setminus S_1$ is more than $3 \cdot 2n/3 = 2n$. Hence, the number of edges incident on nodes in $V \setminus S_1$ is more than n, which contradicts the fact that G is a tree and has at most $n-1$ edges.

The proof in the previous paragraph can be used to show the more general claim that $|S_i|$ is at least one-third the size of $S_i \cup S_{i+1} \cup \cdots \cup S_t$. This means that in each iteration of the **while**-loop, the size of V is decreasing by a third, implying that it takes $O(\log n)$ iterations of the **while**-loop before V becomes empty.

(5.b) The deterministic algorithm for 3-coloring an unoriented tree in $O(\log n)$ rounds, is as follows.

1. Implement the non-distributed algorithm described in the problem in the CONGEST model, but with one change. The given algorithm says that if e has both end points in S, orient it arbitrarily. The change we make is to leave such edges unoriented for now.
   Running time: Each iteration of the **while**-loop takes $O(1)$ rounds in the CONGEST model and therefore, using the proof in 5(a) we see that this algorithm runs in $O(\log n)$ rounds.
2. Use the Cole-Vishkin algorithm to produce, in parallel for all i, $1 \leq i \leq t$, a 3-coloring of $G[S_i]$.
   Running time: $O(\log^* n)$ rounds.
   Note: This is not a 3-coloring of the entire graph because two adjacent nodes, one in S_i and the other in S_j, $j \neq i$, can have the same color.
3. Consider the sets $S_t, S_{t-1}, \ldots, S_2, S_1$ (in this order, one after the other). For each set S_i and for each $j = 1, 2, 3$, let $S_{i,j}$ denote the subset of S_i of nodes colored j. For each set S_i orient every edge e with both endpoints in S_i from the endpoint with larger color to the endpoint with smaller color. In other words, edges with both endpoints in S_i will be oriented from $S_{i,3}$ to $S_{i,2}$ and $S_{i,1}$ and from $S_{i,2}$ to $S_{i,1}$. For each $j = 1, 2, 3$ (considered in this order), process all nodes in $S_{i,j}$ in parallel, as follows. Each node v in $S_{i,j}$, examines the at most two out-neighbors it has and assigns itself a color from $\{1, 2, 3\}$ distinct from the colors assigned to the out-neighbors.
   Running time: $O(t) = O(\log n)$ rounds.

This algorithm runs in $O(\log n)$ rounds. We will now show that it is correct, i.e., it produces a proper 3-coloring of G. The proof is by induction and the inductive hypothesis is the following:

> After set S_i has been processed in Step 2 above, we have constructed a proper 3-coloring of the subgraph of G induced by sets $S_i \cup S_{i+1} \cup \cdots \cup S_t$.

Showing this for $i = 1$ gives us a proper 3-coloring of G.

The inductive hypothesis is trivially true for $i = t+1$. Now suppose that we have processed set S_i and have a proper 3-coloring of the graph induced by $S_i \cup S_{i+1} \cup \cdots \cup S_t$. Now the algorithm processes the set S_{i-1} in three sub-steps: first $S_{i-1,1}$ is processed, then $S_{i-1,2}$ is processed, and then $S_{i-1,3}$ is processed. After set $S_{i-1,1}$ is processed, we are guaranteed that the subgraph induced by $S_{i-1,1} \cup S_i \cup S_{i+1} \cup \cdots \cup S_t$ is properly 3-colored. This is because no two nodes in $S_{i-1,1}$ are adjacent and therefore they can choose colors independently. Furthermore, each node $v \in S_{i-1,1}$ has no out-neighbors in $S_1 \cup S_2 \cup \ldots \cup S_{i-1}$ and at most two neighbors in $S_i \cup S_{i+1} \cup \cdots \cup S_t$ and therefore v can choose a "permanent" color from $\{1, 2, 3\}$ distinct from its neighbors in $S_i \cup S_{i+1} \cup \cdots \cup S_t$. Similarly, after set $S_{i-1,2}$ is processed, we are guaranteed that the subgraph induced by $S_{i-1,2} \cup S_{i-1,1} \cup S_i \cup S_{i+1} \cup \cdots \cup S_t$ has a proper 3-coloring. Note that when a node $v \in S_{i-1,2}$ is processed, any neighbor(s) it has in $S_{i-1,1}$ have already received a "permanent" color and v will take this into account when assigning itself a "permanent" color from $\{1, 2, 3\}$. The same argument holds for $S_{i-1,3}$ and as a result the inductive hypothesis will hold after set S_{i-1} has been processed.

(6) We start the algorithm with nodes exchanging their r-values. This takes $O(1)$ rounds in the CONGEST model. If two neighboring nodes have the same r-values, then the algorithm aborts without producing a coloring. Otherwise, we start the greedy $(\Delta + 1)$-coloring algorithm.

We first show that the probability that two neighboring nodes will have the same r-values is small. Let u and v be two nodes in the network. Then,

$$Pr\Big(r(u) = r(v)\Big) = \frac{1}{2^{c' \cdot \lceil c \log_2 n \rceil}}.$$

By choosing c' to be a large enough constant (e.g., $c' = 3$), we get that $Pr(r(u) = r(v)) < 1/n^3$. In an n-node cycle, there are n pairs of neighboring nodes and using the *union bound* on these n pairs, we get

$$Pr\Big(\text{There exist neighbors } u \text{ and } v\text{: } r(u) = r(v)\Big) < \frac{1}{n^2}.$$

(Make sure you understand this calculation.) We say that there is a *collision* if two neighbors have the same r-values. Thus, $Pr(\text{no collison}) > 1 - 1/n^2$.

We now *condition* the rest of the analysis on the event that there is no collison. For two neighbors v_1 and v_2,

$$Pr\left(r(v_1) > r(v_2) | \text{ no collision}\right) = \frac{1}{2}.$$

This follows from the fact that by symmetry $r(v_1) > r(v_2)$ and $r(v_1) < r(v_2)$ are equally likely. Now consider a path (v_1, v_2, v_3) in the cycle. Then,

$$\begin{aligned} Pr\left(r(v_1) > r(v_2) > r(v_3) | \text{ no collision}\right) &= Pr\left(r(v_1) > r(v_2) | \text{ no collision}\right) \times \\ &\quad\; Pr\left(r(v_2) > r(v_3) | r(v_1) > r(v_2) \text{ and no collision}\right) \\ &= \frac{1}{2} \cdot \frac{1}{3}. \end{aligned}$$

Continuing in this manner, we see that for a path $(v_1, v_2, \ldots, v_t)$ in the cycle

$$Pr\Big(r(v_1) > r(v_2) > \cdots > r(v_t) | \text{ no collision}\Big) = \frac{1}{t!} < \frac{1}{2^{t-1}}.$$

Now let $t = 3\lceil \log_2 n \rceil + 2$ and for this value of t we see that

$$Pr\Big(r(v_1) > r(v_2) > \cdots > r(v_t) |\text{ no collision}\Big) < \frac{1}{2n^3}.$$

We call a path $P = (v_1, v_2, \ldots, v_t)$ in the cycle *decreasing* if $r(v_1) > r(v_2) > \cdots > r(v_t)$. There are $2n$ length-t paths in the cycle and taking a union bound over these we see that

$$Pr\Big(\text{There exists a length-}t = 3\lceil \log_2 n \rceil + 2 \text{ decreasing path} |\text{ no collision}\Big) < \frac{1}{n^2}.$$

Thus conditioned on the "no collisions" event, with probability more than $1 - 1/n^2$, the greedy $(\Delta + 1)$-coloring (which is a 3-coloring since $\Delta = 2$) algorithm will run in at most $3\lceil \log_2 n \rceil + 2 = \Theta(\log n)$ rounds. In other words,

$$Pr\Big(\text{Greedy algorithm produces a 3-coloring in } \Theta(\log n) \text{ rounds} |\text{ no collision}\Big) > 1 - \frac{1}{n^2}.$$

Finally, using the fact that $Pr(A \text{ and } B) = Pr(A|B) \cdot Pr(B)$, we see that the probability that there is no collision *and* the greedy algorithm produces a 3-coloring in at most $t = 3\lceil \log_2 n \rceil + 2$ rounds is more than $(1 - 1/n^2) \cdot (1 - 1/n^2) > 1 - 1/n$.
