ASSIGNMENT 4 SOLUTIONS - DISTRIBUTED ALGORITHMS

(1) A tree can be colored in 2 colors, by assigning the color c_1 to nodes with even depth, and the color c_2 to nodes with odd depth. Yet, this color assignment takes $\Omega(Depth(T))$ time steps ($\Omega(n)$ in the worst case), as proven in class[1].

To get the minimal possible coloring for a tree, we must use *Cole-Wishkin*'s algorithm, as described in class, until we iterate 7 times or get 6 colors or less. Then, we'll peform the shift down algorithm, as described in class, until we get 3 colors or the time counter gets to 7.

The correctness of this algorithm can be proven by combining the correctness proofs of *Cole-Wishkin* and the shift down algorithms, both proved in class.

Time complexity is exactly 7 time steps. Message complexity: on each time step a vertex should know the colors of its parent, using $\Theta(1)$ messages on each edge. Total: $\Theta(|E|) = \Theta(n)$ messages.

(2) We'll use the given algorithm Π to produce 3-coloring of the path p, which contains n vertices. We will build a graph $G = (V, E)$, in which $V = V(p) \cup U$, where $\{u_i | 1 \leq i \leq n\}$, and $E = E(p) \cup \{(v_i, u_i) | v_i \in V\}$. In other words, for each vertex of the original path we'll add a new vertex and connect them by an edge. Note that U is a MIS of the graph G.

Now we'll invoke the Π algorithm on our graph, and receive a 3-coloring of G. If we discard the vertices in U, we get a legal 3-coloring of the original path p.

Time complexity: building G takes $O(n)$ time units, invoking Π takes $Time_n\Pi$ and discarding the extra vertices takes $O(n)$ time units. Total: $O(n + Time_n\Pi)$ time units.

Some of you have proven that $Time(findMIS) \geq Time(3col) = \Omega(log^*(n))$, hence even if $Time_n\Pi = \Theta(1)$ we still can use *Cole-Wishkin*'s algorithm with the same results.

(3) (a) The lemma is correct.
We'll represent the road as a graph $G = (V, E)$ in which the vertices $v_i \in \{v_0, v_1, \ldots, v_{n-1}\}$ represent the gas pumps and an edge $(u_i, u_j), |i - j| = 1$ means that the pumps u and v are connected. Denote by $f(v_i)$ the amount of fuel at the pump v_i, and by $\omega((u_i, u_j) \in E)$ the amount of fuel required to pass the road segment between u_i and u_j.

[1]this can be implemented using distributed BFS from the tree root, for example

Denote by $X(i,j)$ the amount of fuel required to get from u_i to u_j $(u_i, u_j \in V)$ [2]:

$$X(i,j) = \sum_{t=i}^{j-1} (f(u_i) + w(u_i, u_{i+1})) \tag{0.1}$$

It is easy to prove the following:

$$\forall i,j : X(i,j) = -X(j,i) \tag{0.2}$$

$$\forall i : X(i,i) = 0 \tag{0.3}$$

$$\forall i,j,k : X(i,j) = X(i,k) + X(k,j) \tag{0.4}$$

Denote k_{min} as an integer for which $\forall i : X(0, k_{min}) \leq X(0,i)$.

Lemma 0.1. *For each $i \in \{0, 1, \ldots, n-1\}$: $X(k_{min}, i) \geq 0$.*

This means that every gas pump in the road is reachable from the pump $u_{k_{min})}$.

Proof. Assume towards contradiction that there exist an integer t such that $X(k_{min}, t) < 0$. There are two options:

- If u_t is on the path from u_0 to $u_{k_{min}}$, then: $0 = X(0,0) = X(0, k_{min}) + X(k_{min}, 0) = X(0,t) + X(t, k_{min}) + X(k_{min}, 0) \geq X(0, k_{min}) + X(t, k_{min}) + X(k_{min}, 0) = X(t, k_{min})$. Hence $X(t, k_{min}) \leq 0$, so $X(k_{min}, t) \geq 0$, in contradiction to the assumption.
- Otherwise: $0 = X(0,0) = X(0, k_{min}) + X(k_{min}, 0) = X(0, k_{min}) + X(k_{min}, t) + X(t, 0) \leq X(0, k_{min}) + X(k_{min}, t) + X(k_{min}, 0) = X(k_{min}, t)$. Hence $X(k_{min}, t) \leq 0$, in contradiction to the assumption.

□

(b) To find a starting point, we should simply find k_{min}. We can do it by using Broadcast and Convergecast over the ring, in $O(n)$ time units and messages.

(4) A simple solution is to set one vertex color to c_1, then color the rest of the vertices with alternating colors c_2 and c_3, sending messages along the perimeter of the ring. However, this solution requires $O(n)$ time steps which is not the most efficient method.

A better solution is to use *Cole-Wishkin*'s algorithm that colors a general graph with maximal degree of Δ in $\Delta + 1$ colors. This algorithm takes $O(log^* n + \Delta^\Delta)$ time units, and in the case of a ring with $\Delta = 2$, it performs a 3-coloring in $O(log^* n)$ time units.

[2] Note that all the calculations are in $modn$

(5) *Proof.* For each v and p: Let x be the absolute time unit in which the vertex v enters the pulse p. In this time unit, each vertex $u \in V$ holds: $p_u \leq p - s_{max}$.

In the absolute time $x + Time_{Pulse}$, all the vertices $u \in V$ should hold $p_u \leq p - s_{max} + 1$, and each $Time_{Pulse}$ time steps will increment this value by one.

Hence, in time $x + Time_{Pulse} \cdot (s_{max} + 1)$, all the vertices $u \in V$ - among them the vertex v, will hold $p_u \leq p + 1$.

Hence the maximal amount of time steps a vertex can stay in the same pulse number is: $Time_{Gap} \leq Time_{Pulse} \cdot (s_{max} + 1)$. □