

COSC-373: HOMEWORK 3

LEE JIAEN, HYERY YOO, AND ALEXANDER LEE

- Question 1.** (a) We first use the LE/BFS protocol from class and Diameter protocol from HRW 2 to find the leader, BFS tree, and Diameter (D).
(b) Each node then does the following:

```
1 local_counter  $\leftarrow t_v$ 
2 second_counter  $\leftarrow 0$ 
3 status  $\leftarrow inactive$ 
4 reset_cntdown  $\leftarrow D$ 
5 foreach round  $i \in \{0, 1, \dots, d\}$  do
6   if  $i = 0$  then
7     leader sets status = active
8     leader sends reset_cntdown - 1 as a message to its neighbors
9   if received messages then
10    set reset_cntdown = message
11    set status = active
12    send reset_cntdown - 1 as a message to its neighbors
13  if status = active then
14    if second_counter = reset_cntdown then
15      set local_counter = 0
16      set status = inactive
17    else
18      second_counter ++
19  local_counter ++
20 return
```

(Correctness) The protocol above goes through the BFS tree starting from the leader node, which is found by the LE/BFS protocol discussed in class. After using the Diameter protocol in HRW2, every node will know the diameter and know that it will take D rounds to get from the leader to the furthest away node in the BFS tree. Therefore, the leader decides to wait D rounds until resetting and then letting its neighbors know to wait one less round and then those nodes will let their neighbors know to wait one less round and so on. The message that a node receives, tells the node how many more rounds it will take until we reach the end of the BFS tree. Furthermore, *second_counter* and *reset_cntdown* help each node to keep track of when to reset and *second_counter* only start incrementing after the node has received a message and updated its *status* to active.

In $r_0 = \text{round } D$, we would have reached the every node in the BFS tree and the furthest node/nodes from the leader in the BFS tree will be told to wait 0 rounds to reset. Those nodes will then have $\text{second_counter} = \text{reset_cntdown}$ and immediately reset local_counter to 0. In the same round, every other node in the BFS tree would also have $\text{second_counter} = \text{reset_cntdown}$ and reset local_counter to 0. BFS, ensures that we reach every node in the graph. Thus, the above protocol is correct because all nodes would reset local_counter to 0 in $r_0 = \text{round } D$.

(Runtime) All nodes reset after the message reaches the last furthest node in the BFS. The diameter D of a BFS tree is the distance from the root or leader node to the furthest node of the BFS tree. Therefore, it will take at least D rounds for the leader to notify the furthest away node in the BFS tree that they would like to reset. If we have less than D rounds, then the furthest away nodes would not be notified of the reset and fail to reset at the same time as the other nodes in the BFS tree. Thus, this will take $O(D)$ rounds. We also know that the LE/BFS and Diameter algorithm will take also $O(D)$ rounds. Therefore, this protocol will take $O(D)$ rounds.

(Congestion) We already know that the CONGEST protocol BE/LFS from class and Diameter protocol from HRW2 send less than $O(n \log n)$ bits. Furthermore, the protocol above has nodes only send messages containing a constant that indicates how many more rounds until a reset, which only require $O(1)$ bits. Therefore, the protocol above is a CONGEST protocol.

Question 2. TODO

Question 3. (a) We proceed with devising a greedy algorithm that is guaranteed to produce a proper $\Delta + 1$ coloring of G . The algorithm executes as follows:

- (1) Initialize an empty set C , which represents the set of colors.
- (2) Assign any vertex in V the color 1 and add 1 to C .
- (3) Choose an uncolored vertex $v \in V$. Assign v the smallest color in C that has not been assigned to v 's colored neighbors. If v 's colored neighbors have been assigned all the colors in C , add a new color $\max\{C\} + 1$ to C and assign it to v .
- (4) Repeat step (2) until all vertices are colored.

Since the algorithm assigns an uncolored vertex $v \in V$ with the lowest color in C that has not been assigned to v 's neighbors or assigns v with a new color not yet in C , no two neighboring vertices are assigned the same color. Furthermore, because the algorithm only adds a new color to C when all neighbors of v have been assigned all the colors in C and v has at most Δ neighbors, the algorithm adds at most $\Delta + 1$ colors to C . Thus, only $\Delta + 1$ colors are required. Therefore, the algorithm produces a proper $\Delta + 1$ coloring of G .

- (b) Consider the graph G_k of $\Delta + 1$ vertices, where each vertex is connected to every other vertex. In this case, every vertex has a degree of Δ , so the maximum degree is Δ . Now, suppose towards a contradiction that we have a Δ proper coloring of G_k . Thus, at most Δ colors have been assigned to $\Delta + 1$ vertices of G_k . By the Pigeonhole Principle, at least two vertices in G_k have been assigned the same color. Since every vertex is connected to every other vertex, these two vertices with the same color are neighbors. This is

a contradiction since we assumed that we have a Δ proper coloring of G_k , where no two neighboring vertices are assigned the same color. Therefore, G_k does not admit a proper Δ coloring.

(c) Each phase i :

- If received message “halted” from neighbor in phase $i - 1$
 - Remove neighbor from set of neighbors
- Execute protocol Π
- If node is in the MIS found from Π
 - Output i , send message “halted” to neighbors, and halt

(Correctness) In each phase i , protocol Π finds a MIS in the network of remaining nodes that have not halted. The nodes found in the MIS from phase i output color i . Neighboring nodes will not output the same color since they belong to different maximal independent sets.

(Runtime) After each phase if a node is not in a MIS, then at least one of its neighbors becomes part of an MIS, by the maximality condition. As such, after each phase, a node’s degree decreases by at least one. Since a node has at most Δ neighbors, at most Δ phases are needed so that all its neighbors are assigned a color. Hence, each node requires at most $\Delta + 1$ phases to output a color i . Since each phase runs in $T(n)$ rounds, the protocol thus computes a property $\Delta + 1$ coloring of G in $O((\Delta + 1)T(n)) = O(\Delta T(n) + T(n)) = O(\Delta T(n))$ rounds.

(Congestion) Since the above protocol uses the CONGEST protocol Π and apart from in protocol Π , nodes only send the messages “halted” to neighbors, which only require $O(1)$ bits, the protocol above is therefore a CONGEST protocol.