CS6851: Distributed Algorithms

Assignment 1

Student: Krutarth Patel, ee23b137@smail.iitm.ac.in

Problem 1

(a) Let's revisit the lower bound for 2-coloring paths. We showed a lower bound for a graph with $4t+2$ vertices, by combining two paths of length $2t+1$ with an edge. Can we reduce the number of vertices in the lower bound graph by combining two paths of length t each? How does this affect our lower bound?

(b) Now, let us consider a fixed path on $t-2$ vertices, and we add a vertex l to the left end and a vertex r to the right end to make it a path on t vertices. Note that all IDs are fixed except the IDs of l and r. Can you show that a $t-2$ round LOCAL algorithm will fail to 2-color such a path?

Solution (a)

Consider three paths: $P^{(1)}, P^{(2)}, P^{(3)}$ each of length t. Let $P_i^{(n)}$ denote the color of i^{th} node along the n^{th} path(in any arbitrary but fixed direction) We will consider two cases:

(i) $t = 2k+1$
Let A be an algorithm that can color a path in k rounds.
Using pigeon hole principle, we can always find two paths $P^{(i)}$ and $P^{(j)}$ such that $P_{k+1}^{(i)} = P_{k+1}^{(j)}$.
Construct a path $P^{(4)}$ by joining $P_t^{(i)}$ and $P_t^{(j)}$.
$P_{k+1}^{(4)} = P_{t+k+1}^{(4)} = P_{k+1}^{(i)} = P_{k+1}^{(j)}$ because A is a k round LOCAL algorithm.
This is a contradiction since in a valid 2-coloring of $P^{(4)}$, $P_{k+1}^{(4)} \neq P_{t+k+1}^{(4)}$.
$\therefore$ The lower bound is $k+1$ to color a path of length t when $t = 2k+1$.

(ii) $t = 2k$
Let A be an algorithm that can color a path in $k-1$ rounds.
Using pigeon hole principle, we can always find two paths $P^{(i)}$ and $P^{(j)}$ such that $P_{k+1}^{(i)} = P_{k+1}^{(j)}$.
Construct a path $P^{(4)}$ by joining $P_t^{(i)}$ and $P_t^{(j)}$
$P_{k+1}^{(4)} = P_{t+k+1}^{(4)} = P_{k+1}^{(i)} = P_{k+1}^{(j)}$ because A is a $k-1$ round LOCAL algorithm.
This is a contradiction since in a valid 2-coloring of $P^{(4)}$, $P_{k+1}^{(4)} \neq P_{t+k+1}^{(4)}$.
$\therefore$ The lower bound is k to color a path of length t when $t = 2k$.

In general, the lower bound to color a path of length $2t$ is $\frac{t+1}{2}$.

Problem 2

Problem 3

Let T be a tree with n nodes. The Rake and Compress algorithm is used to heirarchically decompose T into a set of paths and singleton vertices. This is done iteratively where each iteration i consists of a rake step (removing leaves) and a compress step (removing degree-two paths) performed on all nodes of the tree. We continue until there are no vertices left in the tree.

(a) Describe how the rake and compress operations in each iteration can be implemented in the LOCAL model in constant rounds.

(b) Let V_i be the set of nodes removed in iteration i, and let $T_i = V(T) \setminus (V_1 \cup ... \cup V_{i-1})$ be the set of nodes that are not removed until the beginning of iteration i (note that $T_1 = V(T)$). Show that $|V_i| \geq |T_i|/2$.

(c) Prove that the total number of rounds/iterations required to reduce the tree to an empty graph is $O(logn)$.

(d) Using the output of the rake-and-compress decomposition, show that there is a deterministic algorithm in the LOCAL model to compute a 3-coloring of an unoriented tree in $O(logn)$ rounds.

Solution (a)

function DELETE(u)
  for $v \in N(u)$ **do**
    send delete link to v
  end for
end function

function UPDATE(u)
  for $v \in N(u)$ **do**
    if received delete link from v **then**
      $N(u) \leftarrow N(u) \setminus v$
      $deg(u) \leftarrow deg(u) - 1$
    end if
  end for
end function

$i \leftarrow i^{th}$ iteration
$u \leftarrow$ node u
$N(u) \leftarrow$ set of adjacent nodes
$deg(u) \leftarrow |N(u)|$
▷ rake step
if $deg(u) = 1$ **then**
  DELETE(u)
end if
▷ compress step

if $deg(u) = 2$ **then**
 DELETE(u)
end if
UPDATE(u)

Solution (b)

Assume the contrary.

$$\implies |V_i| < \frac{|T_i|}{2}.$$

Since, T_i is a forest,

$$\implies \sum_{u \in T_i} deg(u) = 2(|T_i| - 1)$$

Also,

$$\implies \sum_{u \in T_i} deg(u) \geq |V_i| + 3(|T_i| - |V_i|) \geq 2|T_i|$$

Which is a contradiction.

$$\therefore |V_i| \geq \frac{|T_i|}{2}$$

Solution (c)

Using the previous result,

$$|T_i| - |V_i| = |T_{i+1}| < \frac{|T_i|}{2}$$

By recursing on i we get,

$$|T_{i+1}| < \frac{|T_1|}{2^i}$$

The algorithm ends when $|T_{n+1}| = 0$.

$$\implies 2^n > |T_1|$$

$$\therefore n = \lceil \log |T_1| \rceil$$

The round complexity is $O(logn)$

Solution (d)

$i \leftarrow 1$ ▷ iteration
$N(u) \leftarrow$ set of nodes adjacent to u
$deg(u) \leftarrow |N(u)|$
$deleted \leftarrow false$ ▷ whether deleted at i^{th} iteration
$r \leftarrow 0$ ▷ round number
$S \leftarrow \emptyset$ ▷ adjacent nodes deleted after u
$E \leftarrow \emptyset$ ▷ adjacent nodes deleted in the same round as u

```
color ← ID(u)

while deg(u) ≠ 0 ∧ deleted = false do                    ▷ rake step
    if deg(u) = 1 then
        DELETE
    end if
                                                          ▷ compress step
    if deg(u) = 2 then
        DELETE
    end if
    UPDATE
end while

for i ∈ {1...log(n) + 1} do
    if r = i then
        ColeVishkin
    end if
end for

function DELETE
    deleted ← true
    r ← i
    for v ∈ N(u) do
        send delete link to v
    end for
end function

function UPDATE
    for v ∈ N(u) do
        if received delete link from v then
            deg(u) ← deg(u) − 1
            if r < i then
                S ← S ∪ v
            else
                E ← E ∪ v
            end if
        end if
    end for
end function

function ColeVishkin
    for v ∈ S do
        orient edge from u to v
    end for
    for v ∈ E do
        if u > v then
```

orient edge from u to v
else
orient edge from v to u
end if
end for
Apply the Cole Vishkin algorithm on the oriented graph
end function

A few things to note:

1. $|S| + |E| \leq 2$. Can be deduced from the algorithm.
2. Let $E_i = \{(u, v) : u \in V_i, v \in V(T) \setminus (V_1 \cup ... \cup V_{i-1}), (u, v) \in E(T_i)\}$ The induced subgraph $T[E_i]$ is:
   (a) A forest. Since T is a tree.
   (b) $deg(v, T[E_i]) \leq 2 \forall v \in V(T[E_i])$. Since $|S| + |E| \leq 2$.
   (c) Each connected component is thus either a singleton or a path. Cole Vishkin can be used after orienting each edge in E_i according to the algorithm give above.