Coloring Trees

Upper Bound

Thm: \([\text{Cole, Vishkin 86]}\)

For every \(n\)-vertex oriented tree \(T_n\), there is a LOCAL alg. that 3-color \(T_n\) within \(O(\log^* n)\) rounds.

\[\log^* n = \min \{ i : \log^{(i)} n \leq 2 \}\]

where

\[\log^{(1)} n = \log n\]

and

\[\log^{(2)} n = \log \log n\]

\[\log^{(i+1)} n = \log(\log^{(i)} n)\]
Lower Bounds:

Thm 1: Any alg. that 2-colors an n-vertex oriented path $P_n$ requires $\Omega(n)$ rounds.

Thm 2: Any alg. that 3-colors an n-vertex oriented path $P_n$ requires \(\log_2^{\frac{n}{2}} - 2\) rounds.

[Linial 89, Laurinharju-Suomela '14]
3-coloring algorithm for Oriented Trees

0. Step 1: 6-coloring in $O(\log^* n)$ rounds

2. Step 2: Reducing for 3 colors in $o(1)$ rounds.

6-coloring [Intuition]

Initially $C_v = 1D_v$
thus colors with $\log n$ bit.

Goal: In 1 round, re-color vertices by $\log \log n$-bit colors
Alg. 6-color:

* Root r assigns $C_r = 0$
* Code for non-root vertex $v$:
  1. Initially set $C_v = ID_v$ and send to children.

2. Repeat:
   2.1 $i = \min \{0 | C_v[i] \neq C_{P(v)}[i] \}$
   2.2 $C_v = \{ (i, C_v[i]) \}$
   2.3 Send $C_v$ to children.
**Analysis:**

**Claim [correctness]:**
In each iteration, the 6-color alg.
produces a legal coloring.

**Proof:**
Consider two neighbors \( v, w \),
\( \text{w.l.o.g.} \) \( v = \text{parent}(w) \).

- Let \( I, J \) be the indices picked by \( v, w \) (resp.)
- \( I \neq J \) \( \checkmark \)
- \( I = J \): \( C_v(I) \neq C_w(J) \) \( \checkmark \)
Lemma [runtime]:
within $O(\log^k n)$ iterations, the final coloring consists of 6 colors.

Proof:

$k_0 = \# \text{bits in colors after the } i^{th} \text{ iteration}$

$k_0 = \log n$

$k_{i+1} = \ ?$

$K_{i+1} < K_i$ as long as $k_i \geq 4$
Once $k_0 = 3$ we get 6 colors!
Reduce to 3 colors!

Procedure **Shift-Down**: 

1. Root picks a new color in \( \{0, 1, 2, 3\} \) 
2. Each vertex adopts the color of its parent.

**Lemma:** The shift-Down operation preserves legality of coloring. Moreover, siblings are monochromatic.
Idea: Cancel color 3, 4, 5 one by one using Shift-Down!

From 6-colors to 3-colors:

For \( x \in \{3, 4, 5\} \) do: [cancel color \( x \)]

1. Apply Shift-Down
2. Each vertex with color \( x \) picks a free color in \( \{0, 1, 2, 3\} \)

Lemma: Final coloring is legal.

Why?
Lower Bounds

- n-vertex directed path
- Identifiers set monotonically increasing along the path
  \( 10 \rightarrow 20 \rightarrow 23 \rightarrow 24 \rightarrow \ldots \rightarrow 0 \)

Show: Any deterministic alg. for 3-color requires \( \frac{\log^* n}{2} - 2 \) rounds.
By Contradiction:

* Assume there is alg $A$ that 3-colors the path in $t \leq \frac{\log^* n}{2} - 2$ rounds.

* In $t$-rounds, any node learns a sub-path of length $k = 2^{t+1}$

* $f$ function that maps sub-paths of length $k$ to $\{1, 2, 3\}$
$t = 2$:

$A(87, 29, 11, 46, 32)$

$A(29, 11, 46, 32, 77)$
Coloring function

$B$ is a $k$-ary $q$-coloring function if for any set of identifiers $1 \leq a_1 < a_2 < \cdots < a_k < a_{k+1} \leq n$:

1. $B(a_1, a_2, \ldots, a_k) \in \{1, 2, \ldots, q\}$
2. $B(a_1, a_2, \ldots, a_k) \neq B(a_2, \ldots, a_{k+1})$
Obs: If \( \exists \ \text{a } A \) that 3-colors in \( t < \frac{\log^k n}{2} \) rounds, then:

\[ \exists \ k \text{-ary 3-coloring function } B \]

for \( k = 2t+1 \)

Proof:

For any \( 1 \leq a_1 < a_2 < \ldots < a_k \leq n \) define \( B(a_1, a_2, \ldots, a_k) \) as follows:

Simulate \( A \) on path

\[ a_1 \rightarrow a_2 \rightarrow a_3 \rightarrow \ldots \rightarrow a_t \rightarrow a_{t+1} \rightarrow a_{t+2} \rightarrow \ldots \rightarrow a_k \]

\( B(a_1, a_2, \ldots, a_k) \in \text{color of } a_{t+1} \)
Show that \( B \) is \( k \)-ary 3-colorly.

**Property P2:**

Apply \( \Omega \) on a path \( B(a_2, \ldots, a_{k+1}) \)

\[ a_1 - a_2 - a_3 - a_{k+1} - a_{k+2} - a_k - a_{k+1} \]

\( B(a_1, \ldots, a_k) \)

**Goal:** show that there is no function \( B \) that is \( k \)-ary 3-colorly for \( k < \frac{\log^k n}{2} - 2 \)

Using 2 aux. lemmas:
Lemma 1: There is no 1-ary $q$-colored function with $q < n$.

Proof:
Lemma 2:
If \( \exists k\text{-ary } 2\text{-coloring } B \)
then \( \exists (k-1)\text{-ary } 2^{k-1}\text{-coloring } B' \).

For any \( 1 \leq a_1 < a_2 < \ldots < a_{k-1} \leq n \):
\[
B'(a_1, \ldots, a_{k-1}) = \{ i \in [q] \mid \exists a_k > a_{k-1} \\
B(a_1, a_2, \ldots, a_{k-1}, a_k) = i \} \]

\( \Rightarrow \) property \( \Box \) holds.

Assume that \( B'(a_1, \ldots, a_{k-1}) = B'(a_2, \ldots, a_k) \)
for some \( 1 \leq a_1, \ldots, a_k \leq n \).
let \( q' = B(a_1, \ldots, a_{k-1}, a_k) \)

\( \Rightarrow q' \in B(a_2, \ldots, a_{k-1}) \)

By assumption

\( \Rightarrow q' \in B(a_2, \ldots, a_k) \)

By definition

\( \Rightarrow \exists a_{k'} > a_k \text{ s.t.} \)

\( B(a_2, \ldots, a_k, a_{k'}) = q' \)

Contradiction!
So far:

1. \( t \text{-round Alg } A \Rightarrow k \text{-ary 3-coloring} \)
   \( k = \log^* n - 3 \)

2. No 1-ary \( q \)-coloring with \( q < n \)

3. \( k \)-ary \( q \)-coloring \( B \Rightarrow (k-1) \)-ary \( 2^q \)-coloring \( B' \)
* By contradiction assume \( A \) that runs in \( t \leq \frac{\log^2 n}{2} - 2 \)

\[ \exists k\text{-ary 3-color func} \text{tn } B_0 \]

\[ \exists (k-1)\text{-ary } 2^3 \text{ color func. } B_1 \]

\[ \exists (k-2)\text{-ary } 2^3 \text{ " " } B_2 \]

\[ \exists 1\text{-ary } 2^{\frac{k^3}{k}} \text{ - color func } B_{k^3} \]

\[ \lessdot \]

Contradiction!