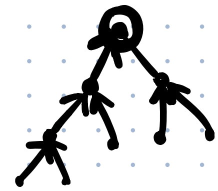
Coloring Trees



Upper Bound

Thm: [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.

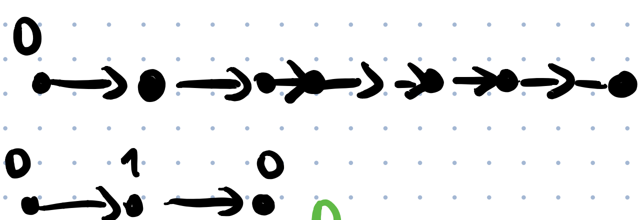
$\log^* n = \min\{i \mid \log^{(i)} n \leq 2\}$ where

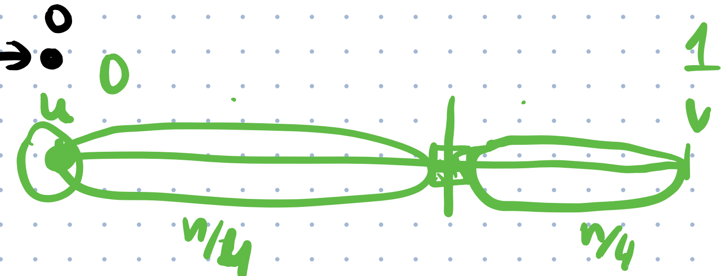
$\log^{(1)} n = \log n$ and $\log^{(2)} n = \log\log n$

$$\log^{(i+1)} n = \log(\log^{(i)} n)$$

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 $\boxed{\frac{\log^* n}{2} - 2}$ rounds.

[Linial 89, Laurinharju-Suomela '14]

3-Coloring algorithm for Oriented Trees

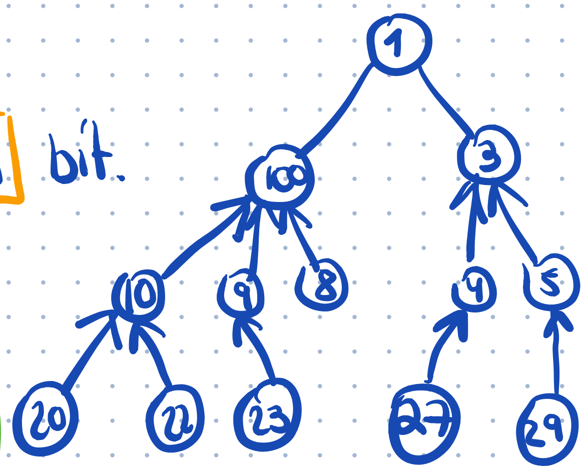
① Step 1: 6-coloring in $O(\log^* n)$ rounds

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

6-coloring [Intuition]

Initially $C_v = ID_v$

thus colors with $\boxed{\log n}$ 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 $\boxed{\boxed{C_v} = ID_v}$ and send to children.

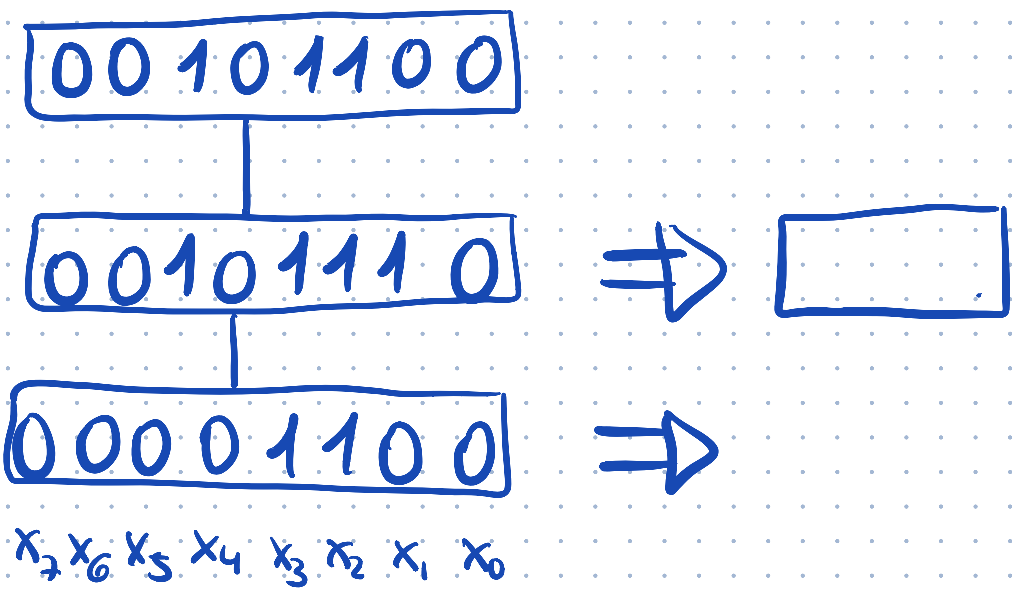
  ② **Repeat:**

  (2.1) $I = \min\{ i \mid C_v[i] \neq C_{p(v)}[i] \}$

  (2.2) $\boxed{C_v = (I, C_v[I])}$

  (2.3) Send C_v to children.

Illustration:



x	Grand-parent	00101110000	$\to$	10010	$\to$	...
y	Parent	10100010000	$\to$	01010	$\to$	111
z	Child	01100010000	$\to$	10001	$\to$	001

Analysis:

Claim[correctness]:

In each iteration, the 6-color alg. produces a legal coloring.

Proof:

Consider two neighbors v, w,

w.l.o.g  $(v = \text{parent}(w))$.

- Let I, J be the indices picked by v, w (resp.).
- $I \neq J$: ✓
- $I = J$: $C_v(I) \neq C_w[J]$ ✓

Lemma [runtime]:

within $O(\log^* n)$ iterations,

the final coloring consists of 6 colors.

Proof:

k_i =: # bits in colors after the i'th iteration.

$k_0 = \log n$

$k_1 = \log n + 1$

$k_2 = \log(\log n + 1) + 1$

$k_{i+1} = ?$

Obs: $k_{i+1} < k_i$ as long as $k_i \geq 4$

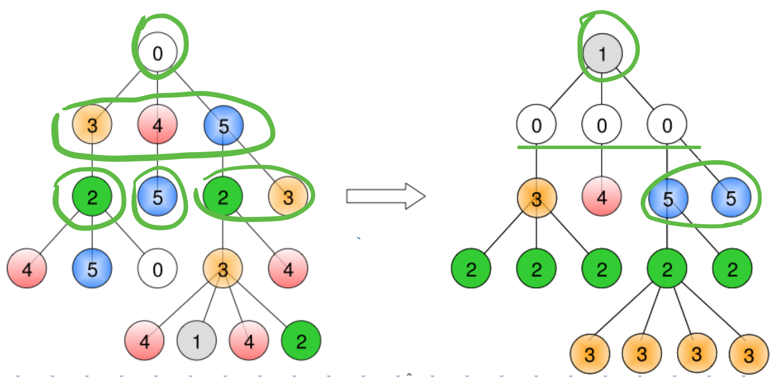
Once $k_i = 3$ we get 6 colors!

0 0 1

Reducing to 3 colors!

Procedure Shift-Down:

1. Root picks ~~new~~ distinct color in $\{0,1,2\}$
2. Each vertex adopts the color of its parent.



<u>Lemma</u>: The shift-Down operation preserves legality of coloring. Moreover, siblings are monochromatic.

Idea: Cancel color $\{\cancel{4}, \cancel{5}, \cancel{6}\}$ $\{3, 4, 5\}$ one by one using Shift-Down!

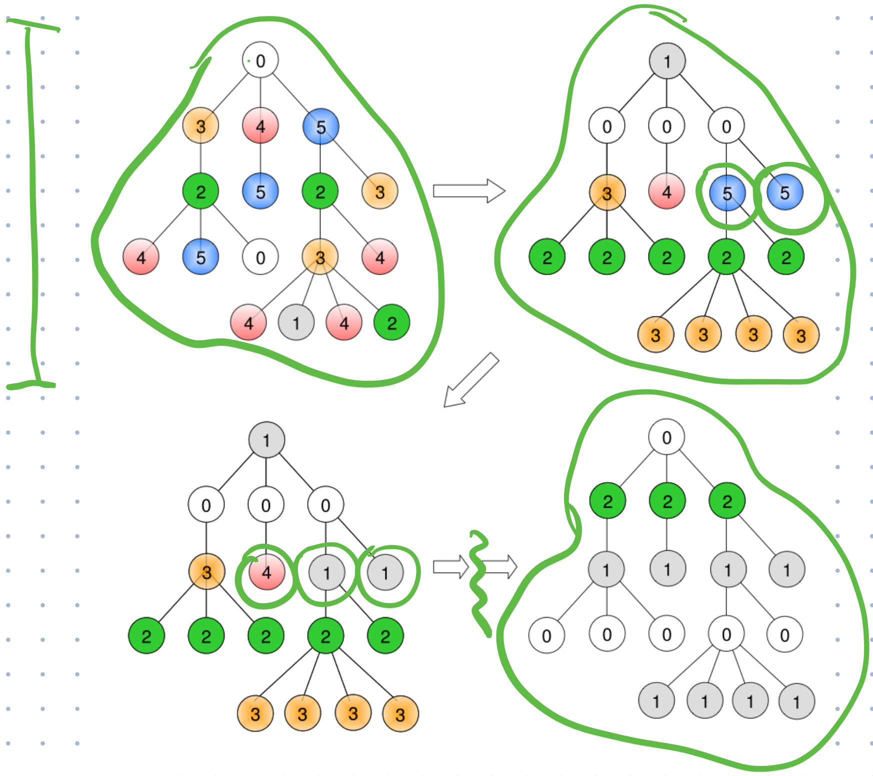
From 6-colors to 3-colors:

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

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

Lemma: Final coloring is legal.

Why?



Lower Bounds

- n-vertex directed path
- Identifiers set monotonically increasing along the path

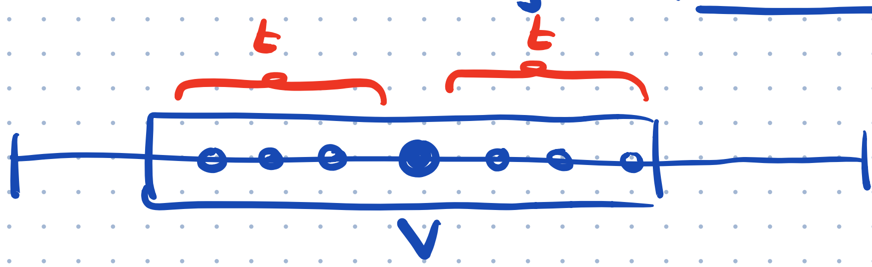
10 → 20 → 23 → 24 → . . . → ○

Show: Any deterministic alg. for 3-color. requires $\boxed{\frac{\log^* n}{2} - 2}$ rounds.

By Contradiction:

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

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



* $\exists$ function that maps sub-paths of length k to $\{1, 2, 3\}$

$\boxed{t=2:}$

A(87, 29, 11, 46, 32)
17
87
29
11
46
32
77
89
17
87
29
11
46
32
77
89
A(29, 11, 46, 32, 77)

Coloring Function

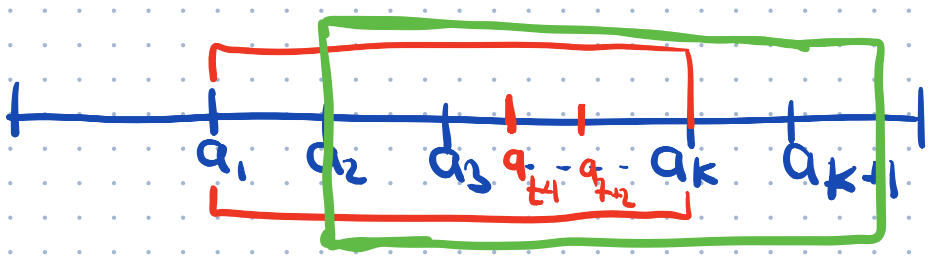
B is k-ary q-coloring function

if for any set of identifiers

$$1 \leq a_1 < a_2 < \cdots < a_k < a_{k+1} \leq n:$$

(P1) $B(a_1, a_2, \ldots, a_k) \in \{1, 2, \ldots, q\}$

(P2) $B(a_1, a_2, \ldots, a_k) \neq B(a_2, \ldots, a_{k+1})$



Obs: If $\exists$ alg. A that 3-colors in $t < \frac{\log^* n}{2} - 2$ rounds, then:

$\exists$ k-ary 3-coloring function B for $\boxed{k = 2t+1}$

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

$$a_1 \to a_2 \to a_3 \to \ldots a_t \to a_{t+1} \to a_{t+2} \to \ldots \to a_{2t+1}$$

$B(a_1, a_2, \ldots, a_k) \equiv$ color of a_{t+1}

Show that B is k-ary 3-coloring.

Property (P2):

Apply $\mathcal{A}$ on a path

$B(a_2, \dots, a_{k+1})$

$$\sim\sim a_1 - a_2 - a_3 \cdots \boxed{a_{t+1}} - \boxed{a_{t+2}} - \cdots a_k - a_{k+1}$$

$B(a_1, \dots, a_k)$

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

Using 2 aux. lemmas:

Lemma 1: There is no 1-ary q-coloring function with $q < n$.

—

Proof:

Lemma 2:

If $\exists$ k-ary q-coloring B
then $\exists$ $(k-1)$-ary 2^q-coloring B'.

For any $1 \leq a_1 < a_2 < \dots < a_{k-1} \leq n$:

$$\boxed{B'(a_1, \dots a_{k-1})} = \{ i \in [q] \mid \exists\, a_k > a_{k-1} \quad B(a_1, a_2, \dots a_{k-1}, a_k) = i \}$$

$\Rightarrow$ property (P1) holds.

Assume that $B'(a_1, \dots, a_{k-1}) = B'(a_2, \dots, a_k)$

for some $1 \leq a_1 \dots\dots a_k \leq n$.

Let $\boxed{q' = B(a_1, \dots, a_{k-1}, a_k)}$

$\Rightarrow \quad q' \in B(a_1, \dots, a_{k-1})$

By assumption

$\Rightarrow \quad q' \in B(a_2, \dots, a_k)$

By def.

$\Rightarrow \quad \exists\, a_{k'} > a_k \quad$ s.t

$$\boxed{B(a_2, \dots, a_k, a_{k'}) = q'}$$

Contradiction!

So far:

① t-round Alg A $\Rightarrow$ k-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 $t \leq \frac{\log^* n}{2} - 2$

$\Rightarrow \exists$ k-ary 3-coloring function B_0

$\Rightarrow \exists$ $(k-1)$-ary 2^3 coloring func. B_1

$\Rightarrow \exists$ $(k-2)$ ary 2^{2^3} " " B_2

$\Rightarrow \exists$ 1-ary $\underbrace{2^{2^{\cdot^{\cdot^{2^3}}}}}_{k}$ - coloring function B_{k-1}

$< n$

Contradiction!