Introduction to the LOCAL SLEEPING Model

Mikaël Rabie

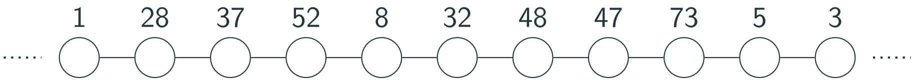
ENEDISC Kickstart Meeting - February 2025

INSTITUT
DE RECHERCHE
EN INFORMATIQUE
FONDAMENTALE

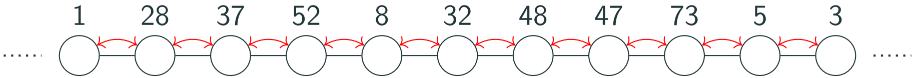
Université de Paris

The LOCAL Model

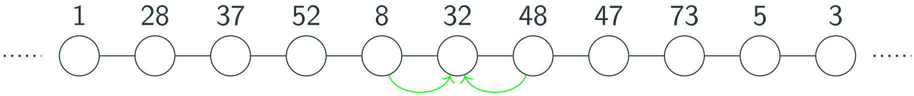
LOCAL Problems on Paths



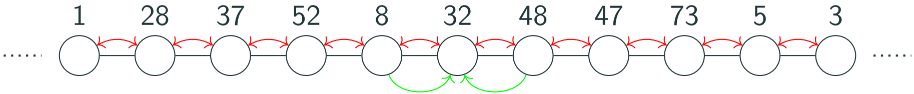
LOCAL Problems on Paths



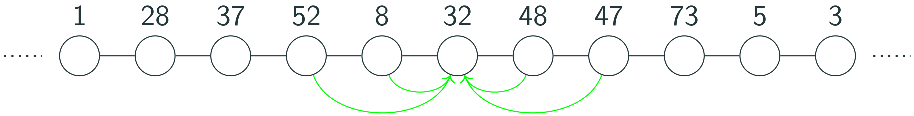
LOCAL Problems on Paths



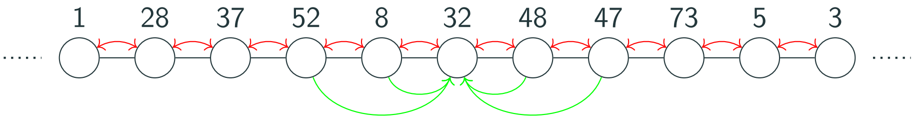
LOCAL Problems on Paths



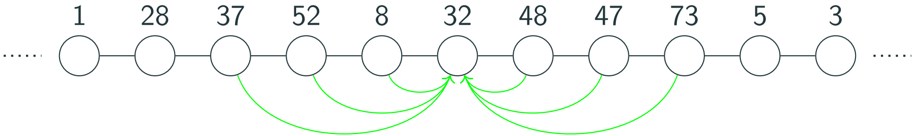
LOCAL Problems on Paths



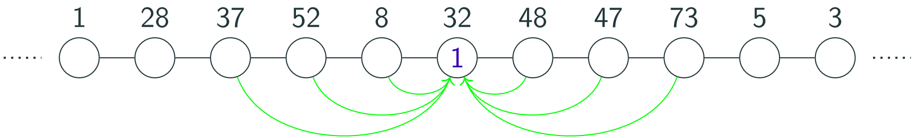
LOCAL Problems on Paths



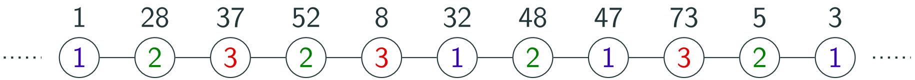
LOCAL Problems on Paths



LOCAL Problems on Paths

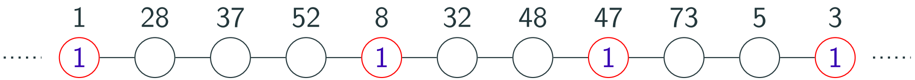


LOCAL Problems on Paths



LOCAL Problems on Paths

LOCAL Problems on Paths



LOCAL Problems on Paths

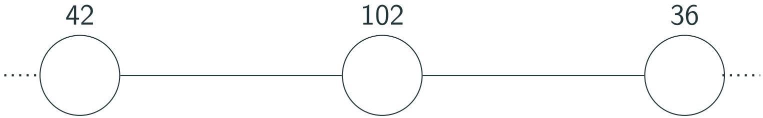
LOCAL Problems on Paths

LOCAL Problems on Paths

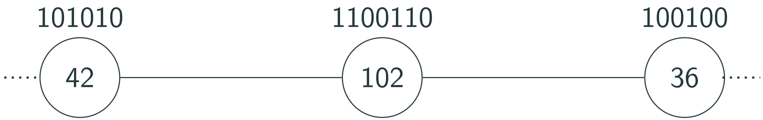
LOCAL Problems on Paths

Cole, Vishkin (1986)
There exists an algorithm to 3-color a path in $O(\log^* n)$ rounds in the LOCAL model.

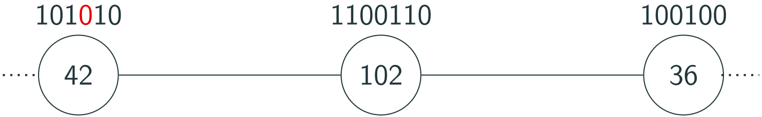
From n colors to $\log n$ colors



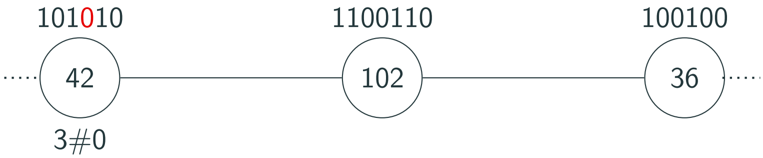
From n colors to $\log n$ colors



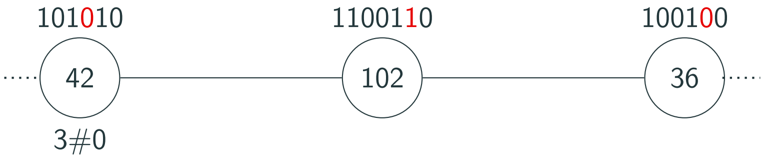
From n colors to $\log n$ colors



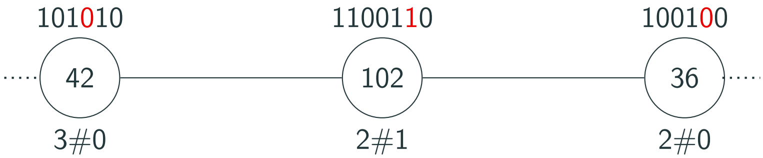
From n colors to $\log n$ colors



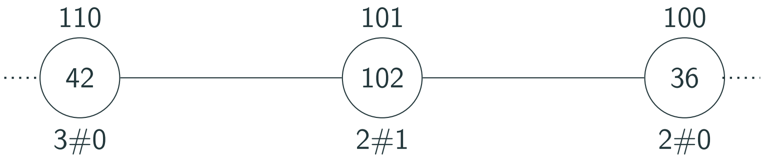
From n colors to $\log n$ colors



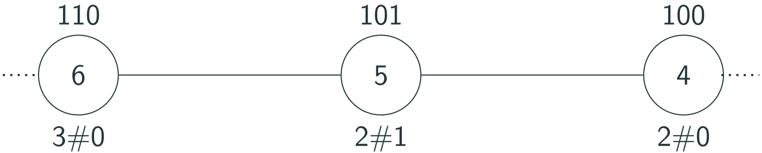
From n colors to $\log n$ colors



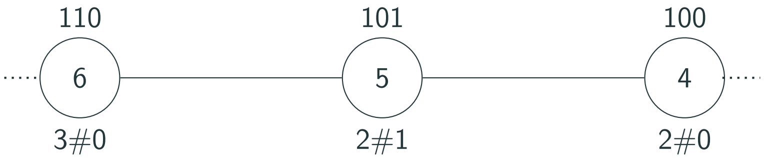
From n colors to $\log n$ colors



From n colors to $\log n$ colors

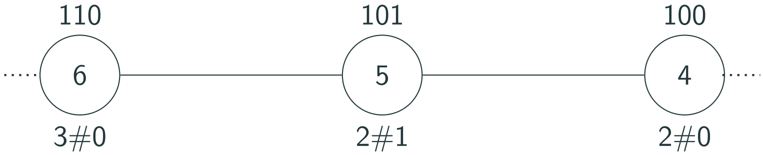


From n colors to $\log n$ colors



n colors $\Rightarrow \log n$ bits $\Rightarrow 2 \log n$ new colors $\Rightarrow \log \log n + 1$ bits

From n colors to $\log n$ colors



n colors $\Rightarrow \log n$ bits $\Rightarrow 2\log n$ new colors $\Rightarrow \log\log n + 1$ bits

After $\log^* n$ iterations, $O(1)$ bits.
After $O(1)$ greedy recoloring steps, 3-coloring.

$(\Delta + 1)$-coloring in the LOCAL model

Linial (1992)

$O(\Delta^2)$-coloring can be computed in $O(\log^* n)$ rounds in the LOCAL model.

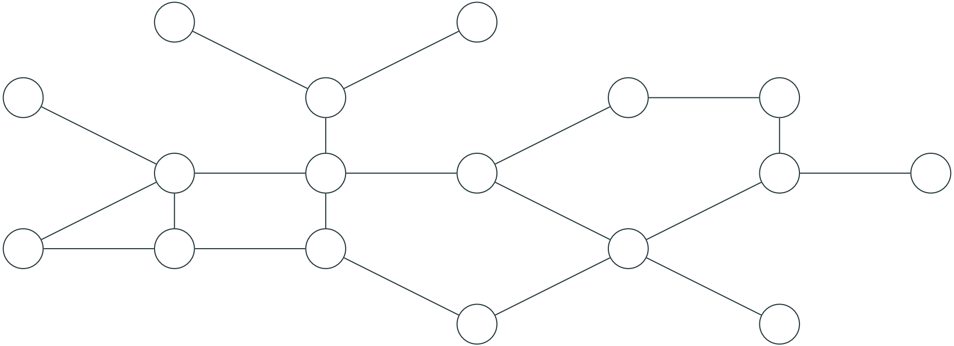
Corollary

$(\Delta + 1)$-coloring can be computed in $O(\log^* n + \Delta^2)$ rounds in the LOCAL model.

The SLEEPING Model

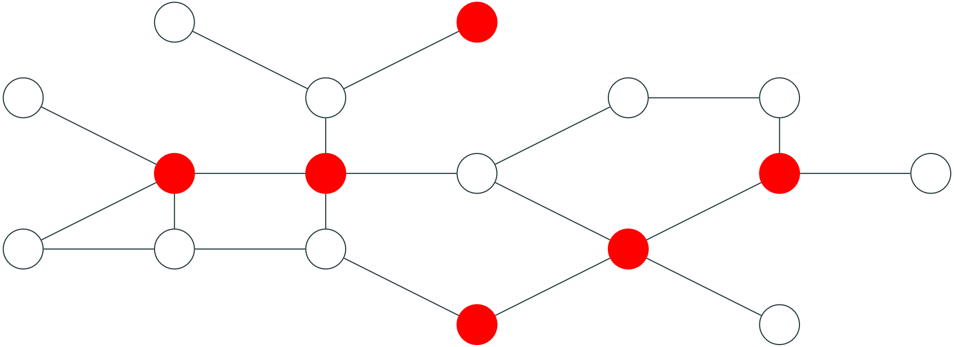
Distributed Sleeping Model

- LOCAL model
- At each round, a node decides if it is active or not
- A node communicates only with its active neighbors
- Complexity : maximal number of awaken rounds for a single node



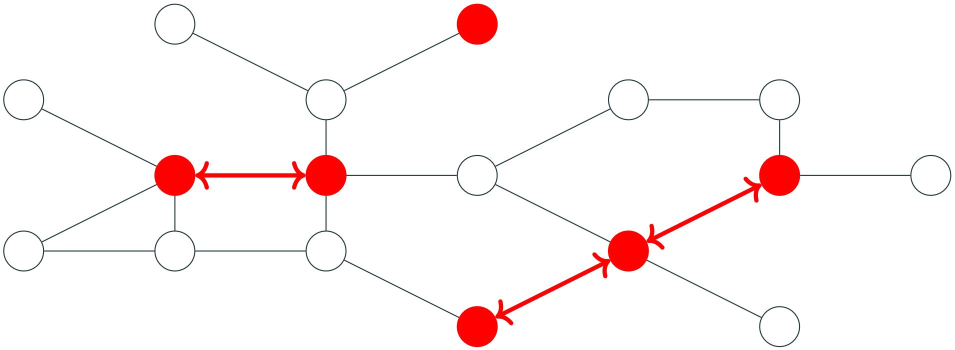
Distributed Sleeping Model

- LOCAL model
- At each round, a node decides if it is active or not
- A node communicates only with its active neighbors
- Complexity : maximal number of awaken rounds for a single node

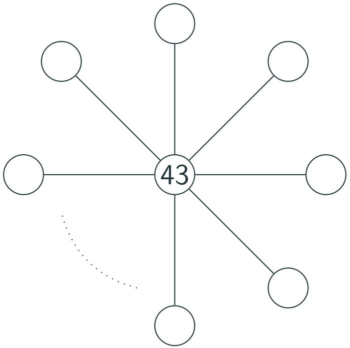


Distributed Sleeping Model

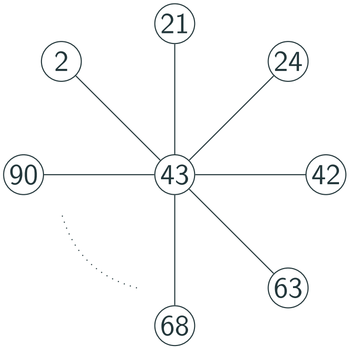
- LOCAL model
- At each round, a node decides if it is active or not
- A node communicates only with its active neighbors
- Complexity : maximal number of awaken rounds for a single node



$\Delta + 1$-Coloring in $O(\Delta)$ awaken rounds

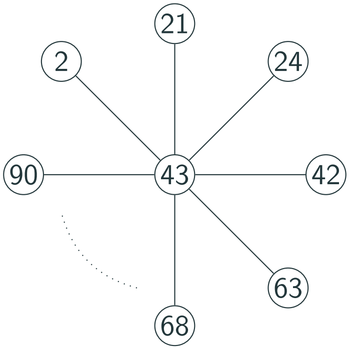


$\Delta + 1$-Coloring in $O(\Delta)$ awaken rounds



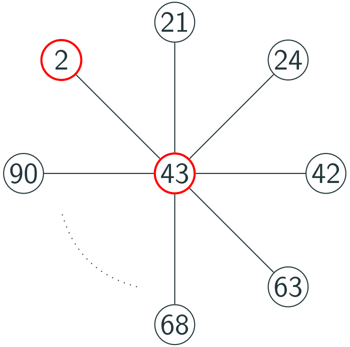
- Round 0 : Learn the identifiers of my neighbors

$\Delta + 1$-Coloring in $O(\Delta)$ awaken rounds

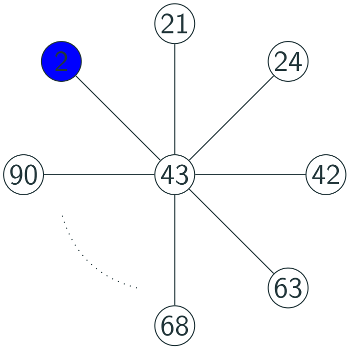


- Round 0 : Learn the identifiers of my neighbors
- For each $i \in N(u)_{\leq 1}$, round i : Wake up

$\Delta + 1$-Coloring in $O(\Delta)$ awaken rounds

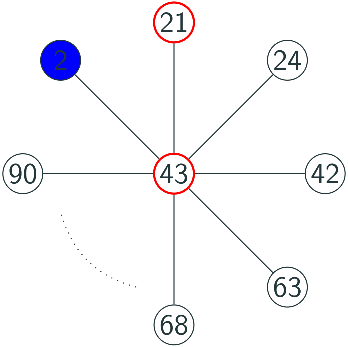


- Round 0 : Learn the identifiers of my neighbors
- For each $i \in N(u)_{\leq 1}$, round i : Wake up
- Round 2 : Node 2 chooses its color



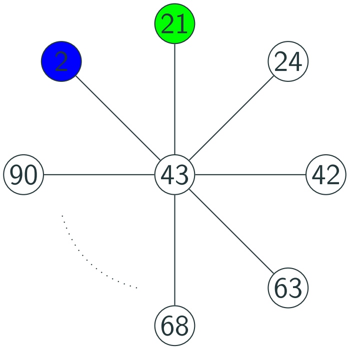
- Round 0 : Learn the identifiers of my neighbors
- For each $i \in N(u)_{\leq 1}$, round i : Wake up
- Round 2 : Node 2 chooses its color

$\Delta + 1$-Coloring in $O(\Delta)$ awaken rounds

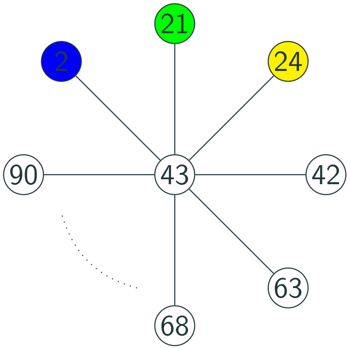


- Round 0 : Learn the identifiers of my neighbors
- For each $i \in N(u)_{\leq 1}$, round i : Wake up
- Round 21 : Node 21 chooses its color

$\Delta + 1$-Coloring in $O(\Delta)$ awaken rounds

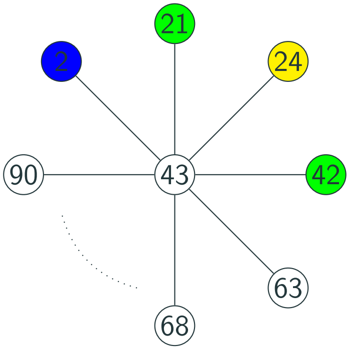


- Round 0 : Learn the identifiers of my neighbors
- For each $i \in N(u)_{\leq 1}$, round i : Wake up
- Round 21 : Node 21 chooses its color



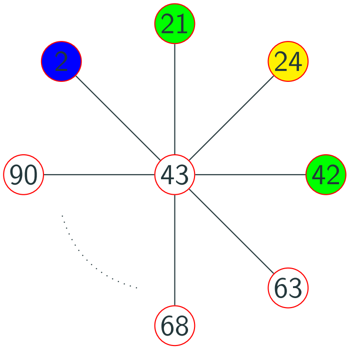
- Round 0 : Learn the identifiers of my neighbors
- For each $i \in N(u)_{\leq 1}$, round i : Wake up
- Round 24 : Node 24 chooses its color

$\Delta + 1$-Coloring in $O(\Delta)$ awaken rounds



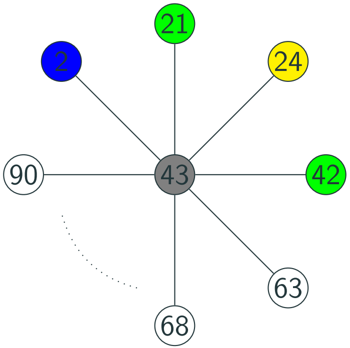
- Round 0 : Learn the identifiers of my neighbors
- For each $i \in N(u)_{\leq 1}$, round i : Wake up
- Round 42 : Node 42 chooses its color

$\Delta + 1$-Coloring in $O(\Delta)$ awaken rounds



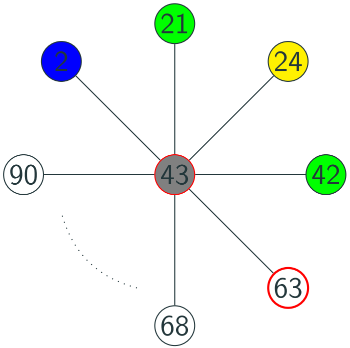
- Round 0 : Learn the identifiers of my neighbors
- For each $i \in N(u)_{\leq 1}$, round i : Wake up
- Round 43 : u learns colors of nodes $2, 21, 24, 42$ and chooses its color

$\Delta + 1$-Coloring in $O(\Delta)$ awaken rounds



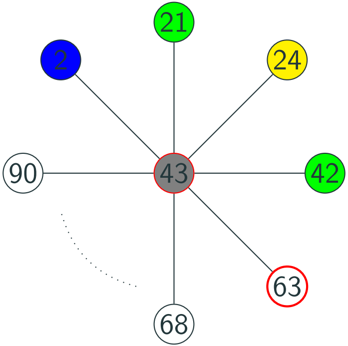
- Round 0 : Learn the identifiers of my neighbors
- For each $i \in N(u)_{\leq 1}$, round i : Wake up
- Round 43 : u learns colors of nodes $2, 21, 24, 42$ and chooses its color

$\Delta + 1$-Coloring in $O(\Delta)$ awaken rounds



- Round 0 : Learn the identifiers of my neighbors
- For each $i \in N(u)_{\leq 1}$, round i : Wake up
- Round 63 : Node 63 learns color of u and chooses its color

$\Delta + 1$-Coloring in $O(\Delta)$ awaken rounds



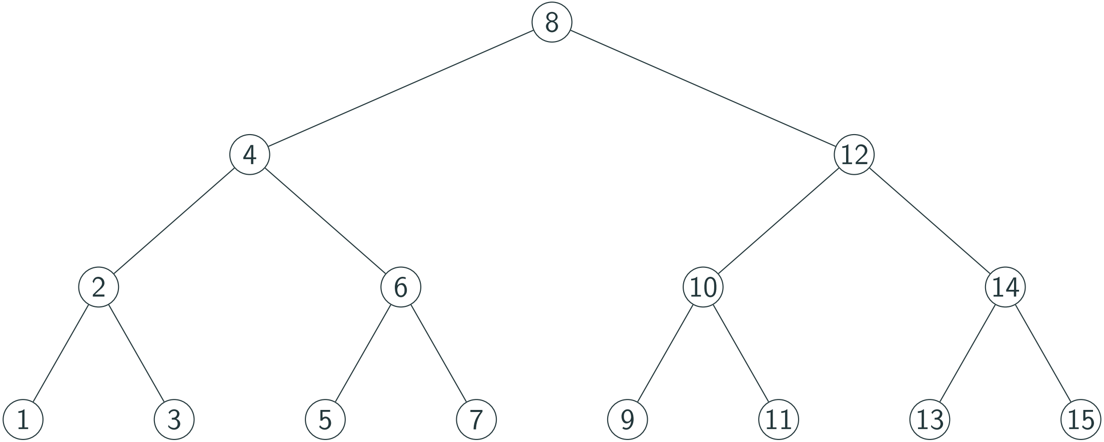
- Round 0 : Learn the identifiers of my neighbors
- For each $i \in N(u)_{\leq 1}$, round i : Wake up
- Round 63 : Node 63 learns color of u and chooses its color

Drawback : The round complexity is $O(M)$, M being the maximal identifier.

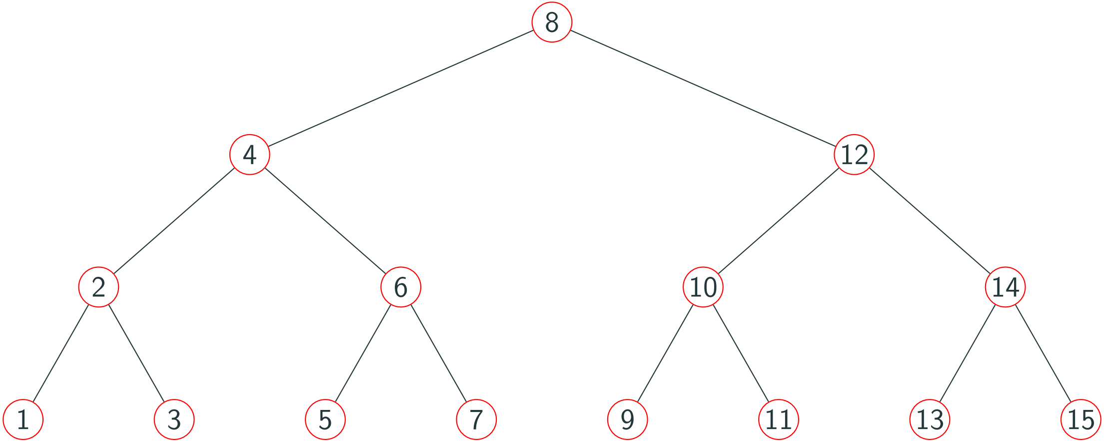
Reduce K colors in $\log K$ awake rounds

Barenboim and Maimon (2021)
Given a K-coloring of the graph, we can compute a $(\Delta + 1)$-coloring in $O(\log K)$ awaken rounds and $O(K)$ rounds in the Sleeping LOCAL model.

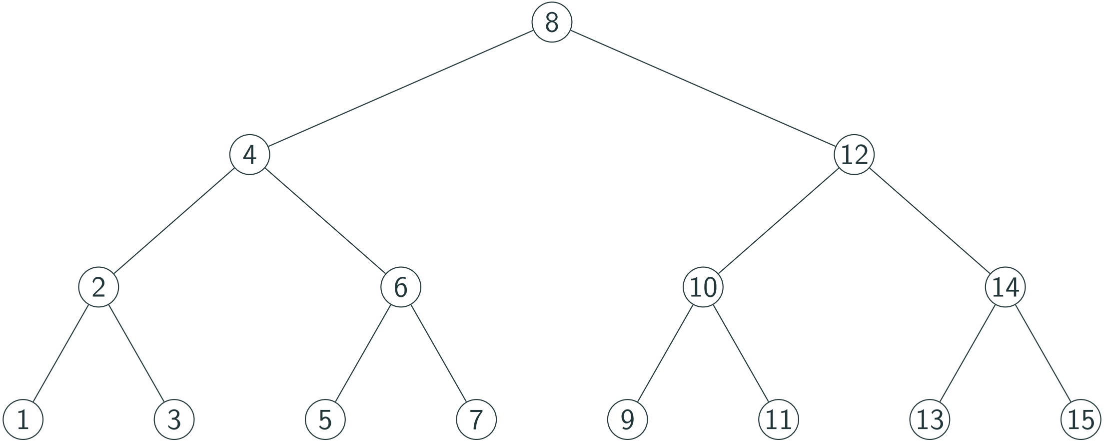
Reduce K colors in $\log K$ awake rounds



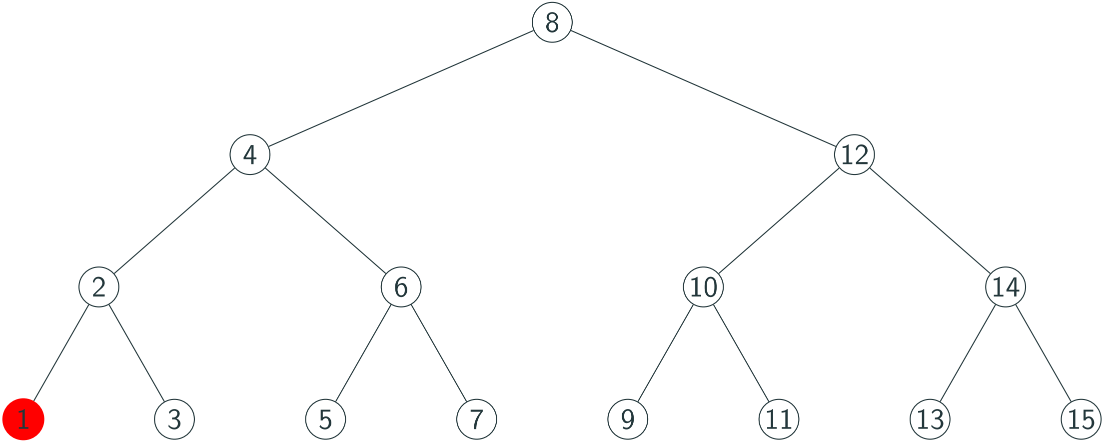
Reduce K colors in $\log K$ awake rounds



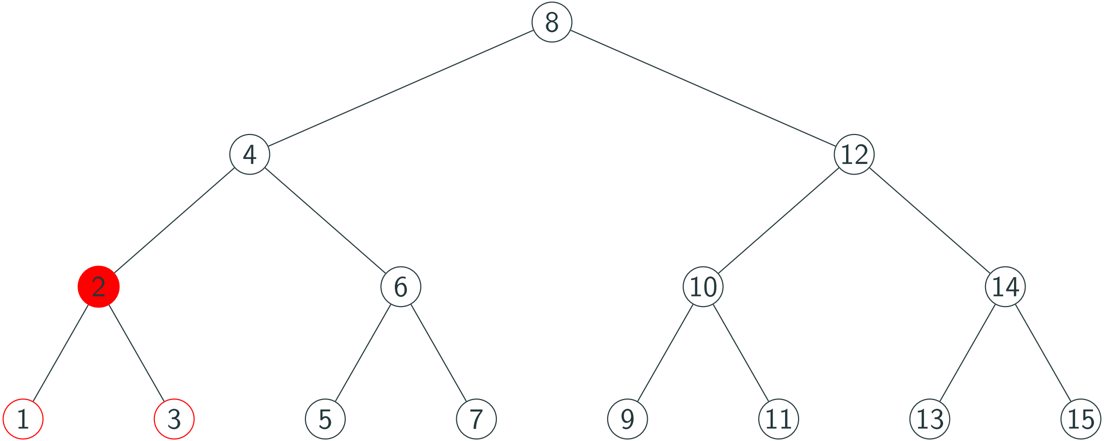
Reduce K colors in $\log K$ awake rounds



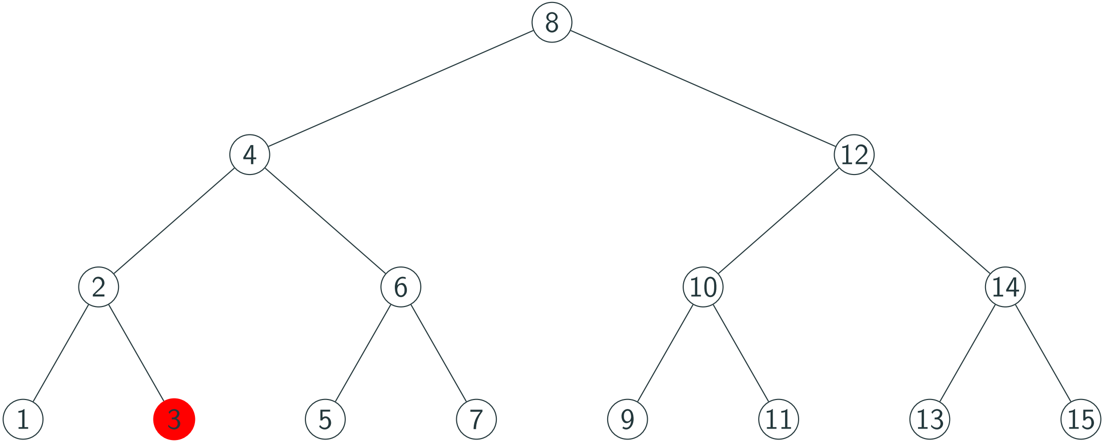
Reduce K colors in $\log K$ awake rounds



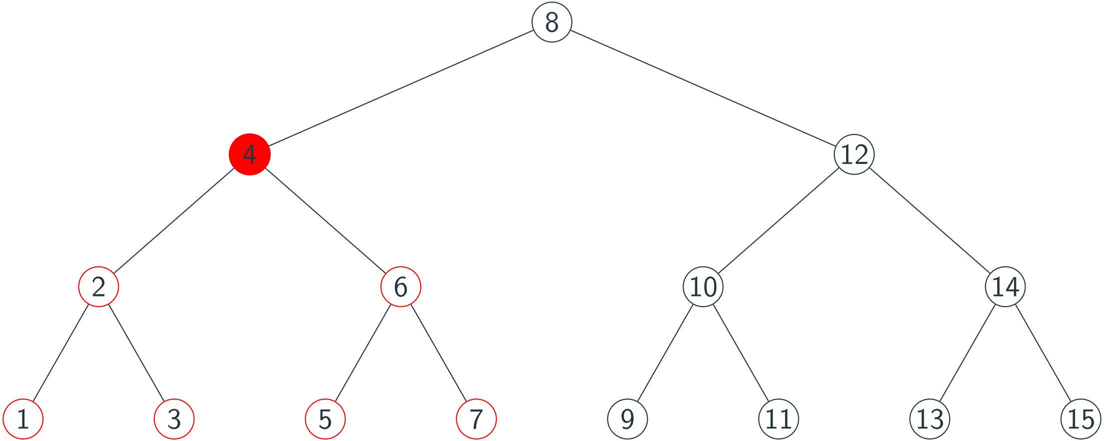
Reduce K colors in $\log K$ awake rounds



Reduce K colors in $\log K$ awake rounds



Reduce K colors in $\log K$ awake rounds



Trade-Off

Find the possible trade-off between awaken and usual rounds to resolve a problem.

$(\Delta + 1)$-coloring of paths :

Awaken rounds	Rounds

Trade-Off

Find the possible trade-off between awaken and usual rounds to resolve a problem.

$(\Delta + 1)$-coloring of paths :

Awaken rounds	Rounds
3	$O(M)$

Trade-Off

Find the possible trade-off between awaken and usual rounds to resolve a problem.

$(\Delta + 1)$-coloring of paths :

Awaken rounds	Rounds
3	$O(M)$
$O(\log^* n)$	$O(\log^* n)$

Trade-Off

Find the possible trade-off between awaken and usual rounds to resolve a problem.

$(\Delta + 1)$-coloring of paths :

Awaken rounds	Rounds
3	$O(M)$
$O(\log^* n)$	$O(\log^* n)$
$3 + k$	$O(\log^k M)$

Trade-Off

Find the possible trade-off between awaken and usual rounds to resolve a problem.

$(\Delta + 1)$-coloring :

Awaken rounds	Rounds
$O(\Delta)$	$O(M)$
$O(\log M)$	$O(M)$
$O(\log^* n + \log \Delta)$	$O(\log^* n + \text{poly}\, \Delta)$

Trade-Off

Find the possible trade-off between awaken and usual rounds to resolve a problem.

$(\Delta + 1)$-coloring :

Awaken rounds	Rounds
$O(\Delta)$	$O(M)$
$O(\log M)$	$O(M)$
$O(\log^* n + \log \Delta)$	$O(\log^* n + \text{poly}\, \Delta)$

Balliu, Fraigniaud, Olivetti, R.

There exists an algorithm that solves $(\Delta + 1)$-coloring with $O(\sqrt{\log n} \cdot \log^* n)$ awake-complexity and round-complexity $\text{poly}(M)$.

Maximal Independent Sets

Dufoulon, Moses, Pandurangan (2023)
Maximal Independent Set :

	Sleeping-Rand-MIS-1	Sleeping-Rand-MIS-2
Node-averaged awake complexity	$O(1)$	$O(1)$
Worst-case awake complexity	$O(\log\log n)$	$O((\log\log n)\log^* n)$
Total round complexity	$O(\text{poly } n)$	$O((\log^3 n)(\log\log n)\ \log^* n)$

The $\log n$ Complexity

Full Knowledge of the Graph

Barenboim and Maimon (2021)
Any graph problem can be solved in $O(\log n)$ awaken rounds in the Sleeping LOCAL model.

This algorithm takes $O(\operatorname{poly} M)$ rounds.

Full Knowledge of the Graph

Barenboim and Maimon (2021)
Any graph problem can be solved in $O(\log n)$ awaken rounds in the Sleeping LOCAL model.

This algorithm takes $O(\text{poly}\, M)$ rounds.

Distributed Layered Tree (DLT) - Oriented Spanning Tree such as :

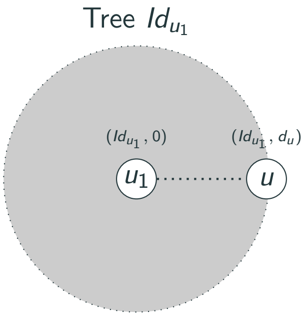
- Each vertex has a label
- The label of a vertex is bigger than its parent's
- Each vertex knows the label of its neighbours in the tree

Constant Coordination
Broadcast and Convergecast can be done in $O(1)$ rounds in a DLT.

Building a DLT

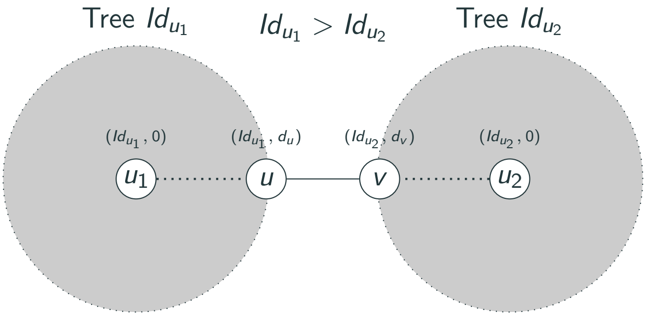
Barenboim and Maimon (2021)
A DLT can be built in $O(\log n)$ awaken rounds in the Sleeping LOCAL model.

Building a DLT



- Labels are of the form (a, b), ordered lexicographically.
- At the beginning, all nodes have label $(Id(u), 0)$.
- At the beginning of each expand step, all nodes of a subtree T are of the form $(L(T), b)$.

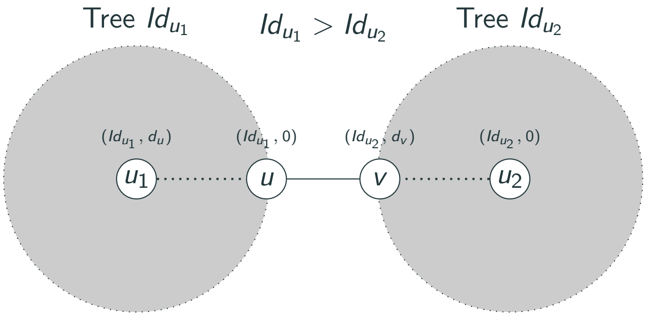
Building a DLT



- Repeat $\log n$ times :

1. Select a neighbour Tree T' with smaller label ($Id_{u_1} > Id_{u2}$).

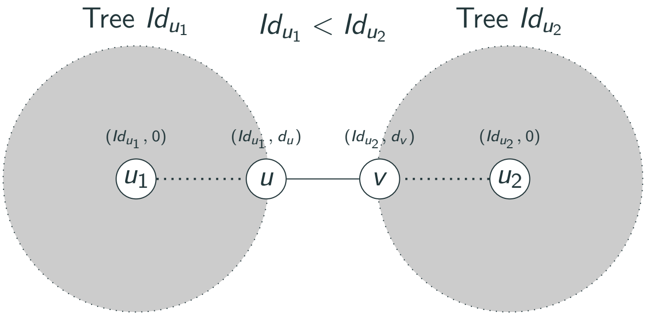
Building a DLT



- Repeat $\log n$ times :

2. Merge T and T', using an edge (u, v).

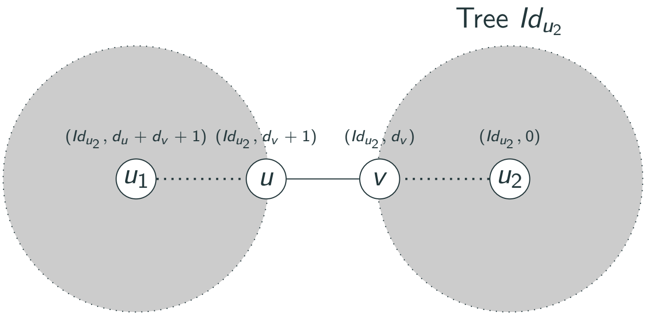
Building a DLT



- Repeat $\log n$ times :

3. If T could not choose a neighbour and was not selected
   T chooses a tree T' to join using an edge (u, v).
   This forms a star of trees around $T' \Rightarrow O(1)$ merge rounds.

Building a DLT



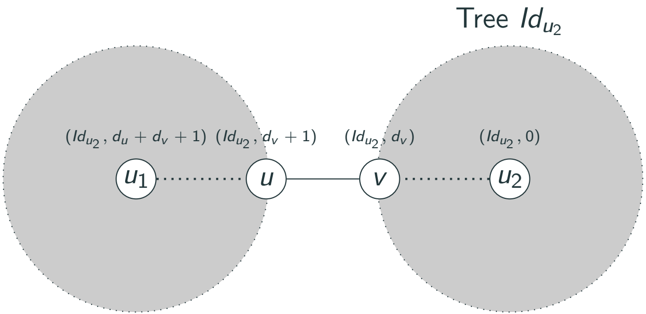
- Repeat $\log n$ times :

3. If T could not choose a neighbour and was not selected

   T chooses a tree T' to join using an edge (u, v).

   This forms a star of trees around $T' \Rightarrow O(1)$ merge rounds.

Building a DLT



- Repeat $\log n$ times :

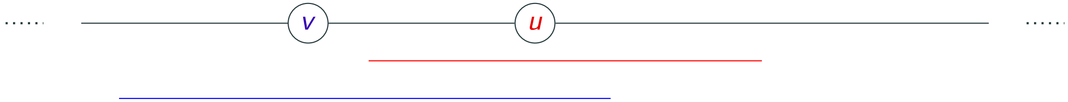
4. All nodes learn their new neighbours in the tree.
5. Convergecast to gather the new structure of the component C to the root r.
6. Broadcast a new labelling $(L(r), dist(r))$.

Sleeping Lower Bound

Augustine *et. al* (2022)
Any algorithm to solve 2-coloring with probability exceeding $1/8$ on a ring network requires $\Omega(\log n)$ awake time.

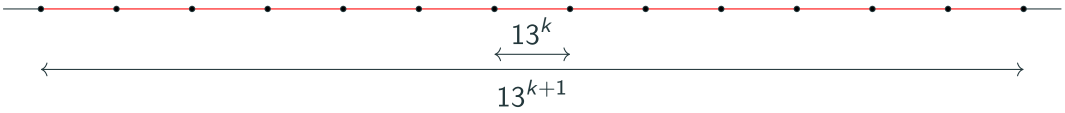
Sleeping Lower Bound

Augustine *et. al* (2022)
Any algorithm to solve 2-coloring with probability exceeding $1/8$ on a ring network requires $\Omega(\log n)$ awake time.



- After k rounds, a node knows about some segment that includes itself
- No node v on the left of u in the path can know more than u on its right

Sleeping Lower Bound



By induction : For any k, for any segment I of 13^k nodes, there exists, with probability $\mathcal{P} > 1/2$, a node $u \in I$ who knows less than I after k rounds.

Sleeping Lower Bound

By induction : For any k, for any segment I of 13^k nodes, there exists, with probability $\mathcal{P} > 1/2$, a node $u \in I$ who knows less than I after k rounds.

- Probability that it is true on 5 of the 13 subsegments is at least $5/6$
- Probability that B, C or D wakes up before A and E is at least $1/2$

Some Questions

- How to adapt known algorithm and what can be new techniques ?
- What are the new complexity classes ?
- How to introduce energy efficiency in other (asynchronous) models ?

Some Questions

- How to adapt known algorithm and what can be new techniques ?
- What are the new complexity classes ?
- How to introduce energy efficiency in other (asynchronous) models ?

Thank You !