

Distributed Algorithms 2022

4 LOCAL model:
Unique identifiers

LOCAL model

=

port-numbering model
+ unique identifiers

Nodes have distinct labels from $\{1, 2, \dots, \text{poly}(n)\}$

LOCAL model

- Everything can be solved in $\text{diam}(G)+1$ rounds!
- Universal algorithm: *“each node tells its neighbors everything it knows”*
 - **1 round:** everyone aware of its adjacent nodes and incident edges
 - **T rounds:** everyone aware of all nodes and edges within distance T
 - **$\text{diam}(G)+1$ rounds:** everyone knows G

LOCAL model

- Not so interesting:

“What can be computed?”

- Very interesting:

“What can be computed efficiently?”

(efficient $\approx o(\text{diam}(G))$ rounds)

Coloring

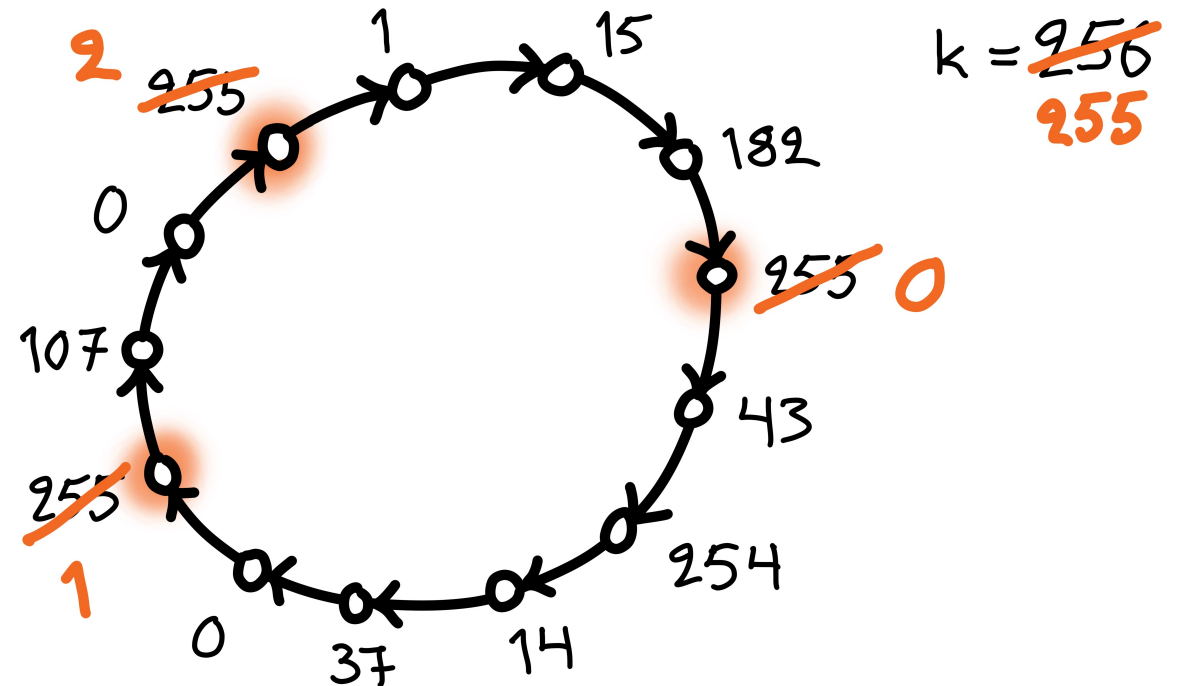
Input	Output	Rounds	Algorithm
Unique IDs	$O(\Delta^2)$ -coloring	$O(\log^* n)$	Cover-free families
$O(\Delta^2)$ -coloring	$O(\Delta)$ -coloring	$O(\Delta)$	Rotating clocks
$O(\Delta)$ -coloring	$(\Delta+1)$ -coloring	$O(\Delta)$	Greedy color reduction
Unique IDs	$(\Delta+1)$ -coloring	$O(\Delta + \log^* n)$	Combine these algorithms

Coloring

Input	Output	Rounds	Algorithm
Unique IDs	$O(\Delta^2)$ -coloring	$O(\log^* n)$	Cover-free families
$O(\Delta^2)$ -coloring	$O(\Delta)$ -coloring	$O(\Delta)$	Rotating clocks
$O(\Delta)$ -coloring	$(\Delta+1)$ -coloring	$O(\Delta)$	Greedy color reduction
Unique IDs	$(\Delta+1)$ -coloring	$O(\Delta + \log^* n)$	Combine these algorithms

Greedy color reduction

- If I am a *local maximum*:
 - pick the smallest free color that is not used by any of my neighbors
- $k+1$ colors $\rightarrow k$ colors
 - provided that $k \geq \Delta+1$

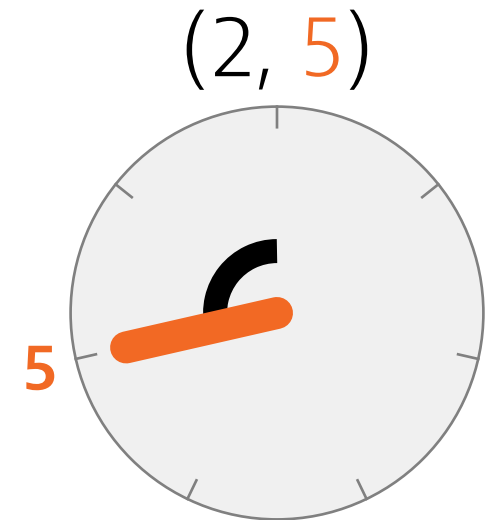
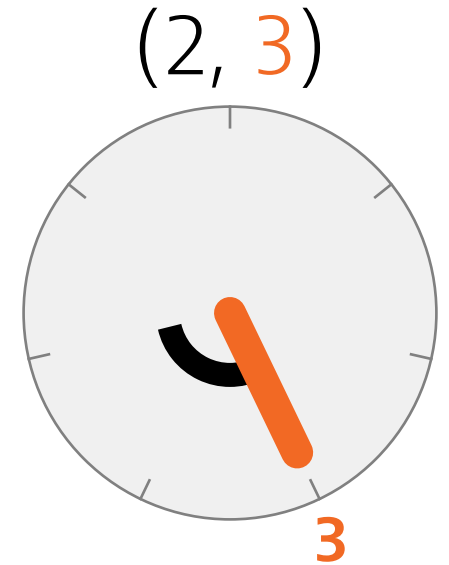


Coloring

Input	Output	Rounds	Algorithm
Unique IDs	$O(\Delta^2)$ -coloring	$O(\log^* n)$	Cover-free families
$O(\Delta^2)$ -coloring	$O(\Delta)$ -coloring	$O(\Delta)$	Rotating clocks
$O(\Delta)$ -coloring	$(\Delta+1)$ -coloring	$O(\Delta)$	Greedy color reduction
Unique IDs	$(\Delta+1)$ -coloring	$O(\Delta + \log^* n)$	Combine these algorithms

Rotating clocks

- $q = \text{prime}, q > 2\Delta$
- q^2 colors $\rightarrow q$ colors in q rounds
- If no conflicts:
 - $(a, b) \rightarrow (0, b)$
- Otherwise:
 - $(a, b) \rightarrow (a, b+a \bmod q)$



Coloring

Input	Output	Rounds	Algorithm
Unique IDs	$O(\Delta^2)$ -coloring	$O(\log^* n)$	Cover-free families
$O(\Delta^2)$ -coloring	$O(\Delta)$ -coloring	$O(\Delta)$	Rotating clocks
$O(\Delta)$ -coloring	$(\Delta+1)$ -coloring	$O(\Delta)$	Greedy color reduction
Unique IDs	$(\Delta+1)$ -coloring	$O(\Delta + \log^* n)$	Combine these algorithms

Cover-free families

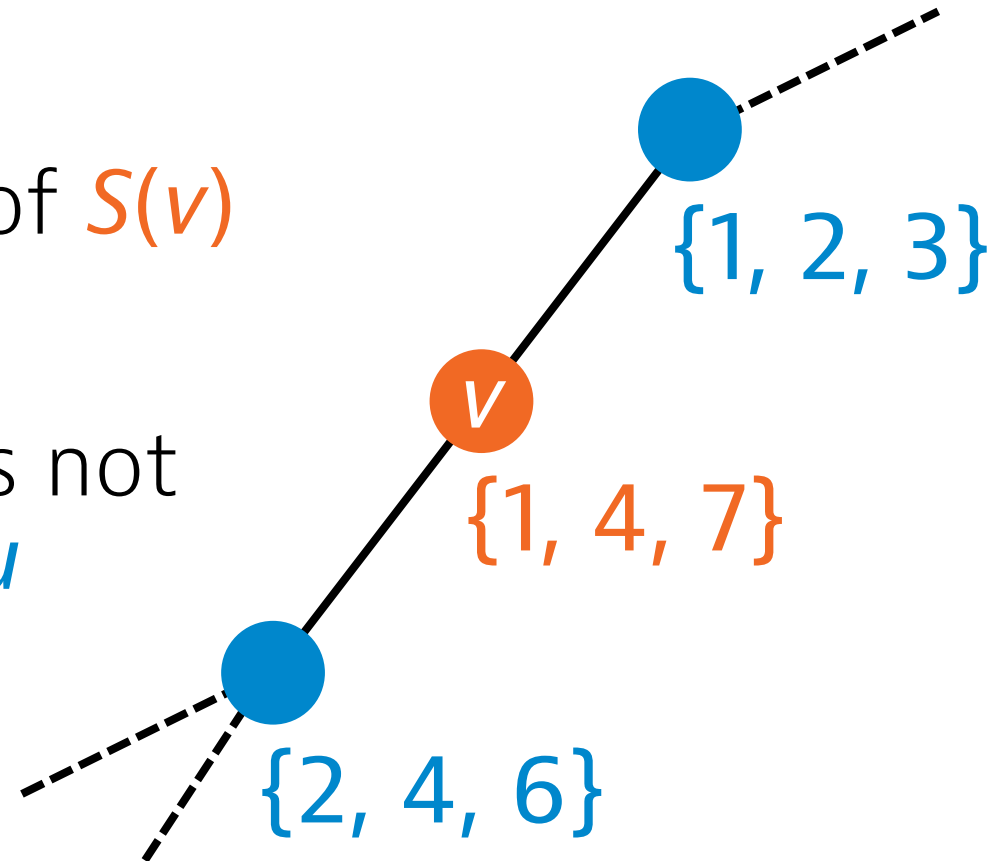
- Old color of node v is a set $S(v) \subseteq \{1, 2, \dots, m\}$

- **Promise:**

- new color of v is an element of $S(v)$

- **Safe:**

- pick an element of $S(v)$ that is not in any $S(u)$ for any neighbor u

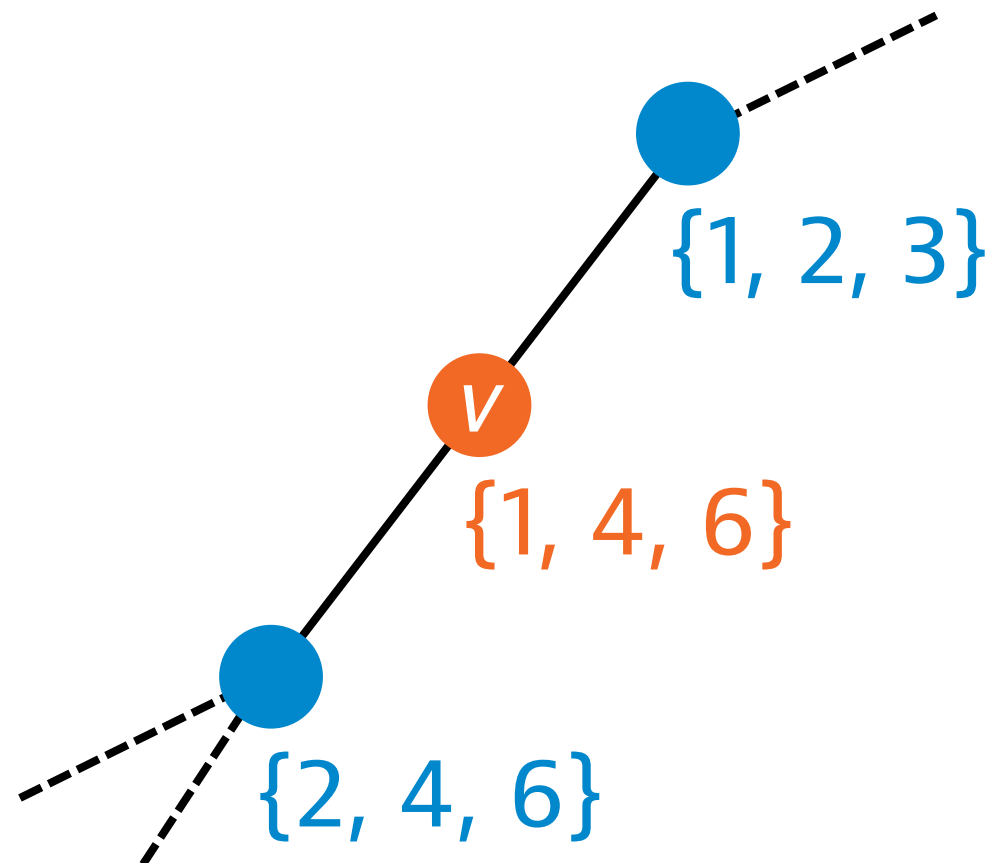


Cover-free families

- Old color of node v is a set $S(v) \subseteq \{1, 2, \dots, m\}$

- **Bad:**

- sets of neighbors cover all values in my set
- no safe choice left

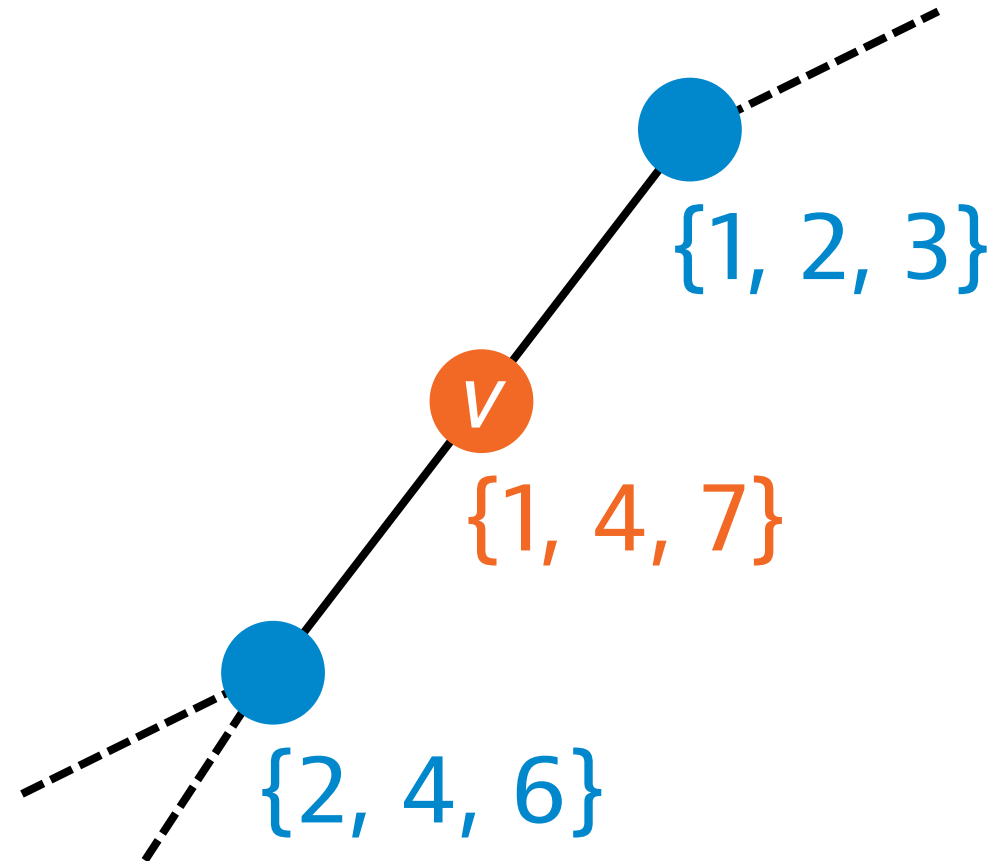


Cover-free families

- Old color of node v is a set $S(v) \subseteq \{1, 2, \dots, m\}$

- **Good:**

- my set is not fully covered by my neighbors
- there is a safe choice



1-cover-free family

- For up to 1 neighbor these sets are good:

$$S_1 = \{ 1, 2 \}$$

$$S_2 = \{ 1, 3 \}$$

$$S_3 = \{ 1, 4 \}$$

$$S_4 = \{ 2, 3 \}$$

$$S_5 = \{ 2, 4 \}$$

$$S_6 = \{ 3, 4 \}$$

2-cover-free family

- For up to 2 neighbors these sets are good:

$$S_1 = \{ 1, 2, 3 \}$$

$$S_2 = \{ \quad 3, 4, 5 \}$$

$$S_3 = \{ \quad \quad 5, 6, 7 \}$$

$$S_4 = \{ 1, \quad 4, \quad 7 \}$$

$$S_5 = \{ \quad 2, \quad 4, \quad 6 \}$$

Cover-free families

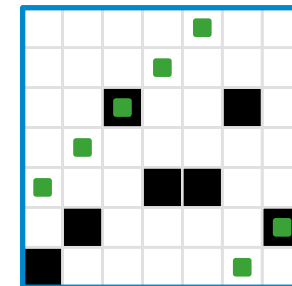
- Assume: x -coloring, maximum degree Δ
- Assume: a Δ -cover-free family S_1, S_2, \dots, S_x
 - all subsets of $\{1, 2, \dots, m\}$
- Nodes of color c pick set S_c
- There is always a safe choice for any node!
- Color reduction from x to m

Cover-free families

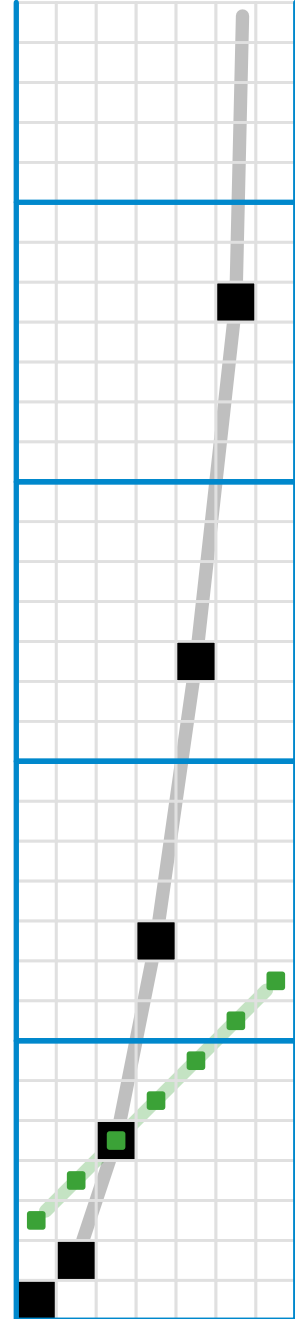
- Δ -cover-free family S_1, S_2, \dots, S_x
 - all subsets of $\{1, 2, \dots, m\}$
- Good if:
 - Δ large \rightarrow works in high-degree graphs
 - x large \rightarrow tolerates many input colors
 - m small \rightarrow produces a good output coloring
- E.g. $x = m$ is trivial (why?)

**Constructing
cover-free
families**

- $q = \text{prime}$, **$\text{GF}(q) \approx \text{integers modulo } q$**
- $f = \text{degree-}d$ **polynomial** over $\text{GF}(q)$
 - at most d points where $f(x) = g(x)$
 - q^{d+1} possible polynomials
- **$S_f = \{ (x, f(x)) \mid x = 0, 1, \dots, q-1 \}$**
 - base set: all q^2 possible pairs (x, y)
 - q^{d+1} possible subsets, each with q elements
 - intersection of S_f and S_g has size at most d
- If $q > \Delta d$: a Δ -cover-free family
 - why?



$f(x), g(x) \bmod q$



$f(x), g(x)$

Cover-free families

- Construct Δ -cover-free families with suitable parameters

- $n \rightarrow \approx \Delta^2 \log^2 n$
 $\rightarrow \approx \Delta^2 \log^2 \log n$
 $\rightarrow \approx \Delta^2 \log^2 \log \log n$
 \dots
 $\rightarrow \approx \Delta^2 \log^2 \Delta$
 $\rightarrow \approx \Delta^2$

} $O(\log^* n)$ steps

Coloring

Input	Output	Rounds	Algorithm
Unique IDs	$O(\Delta^2)$ -coloring	$O(\log^* n)$	Cover-free families
$O(\Delta^2)$ -coloring	$O(\Delta)$ -coloring	$O(\Delta)$	Rotating clocks
$O(\Delta)$ -coloring	$(\Delta+1)$ -coloring	$O(\Delta)$	Greedy color reduction
Unique IDs	$(\Delta+1)$ -coloring	$O(\Delta + \log^* n)$	Combine these algorithms