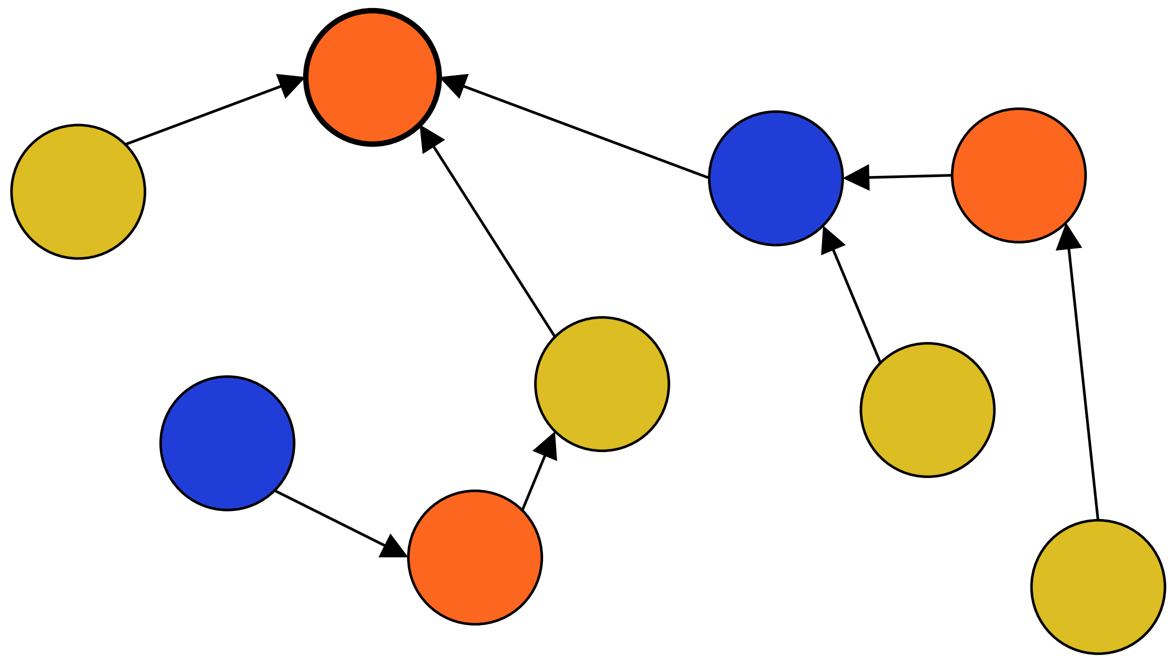
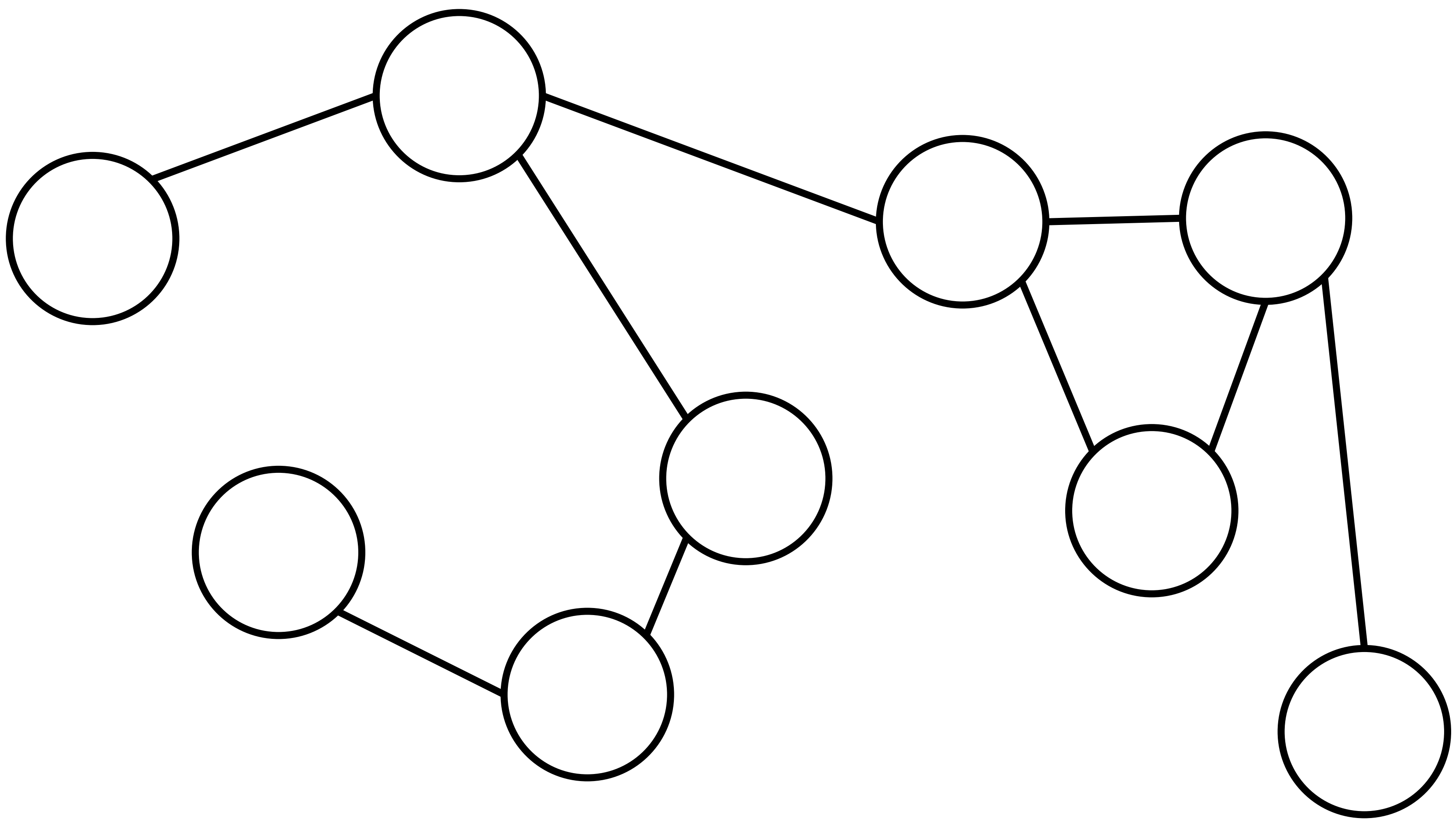
Vertex Coloring in the LOCAL model



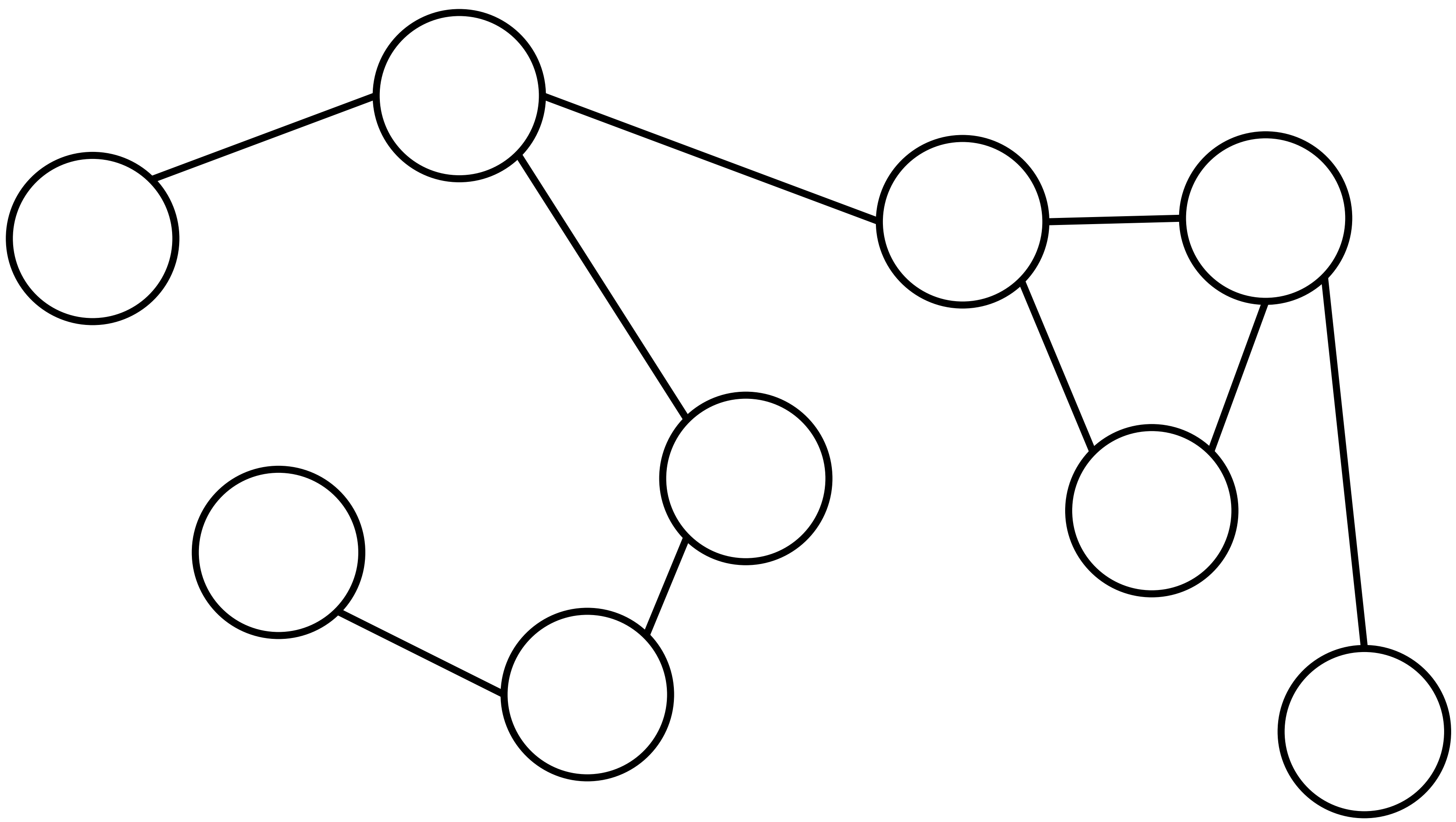
Case Study: Vertex Coloring



vertex coloring:

assign a color to each node of the graph such neighbors have different colors

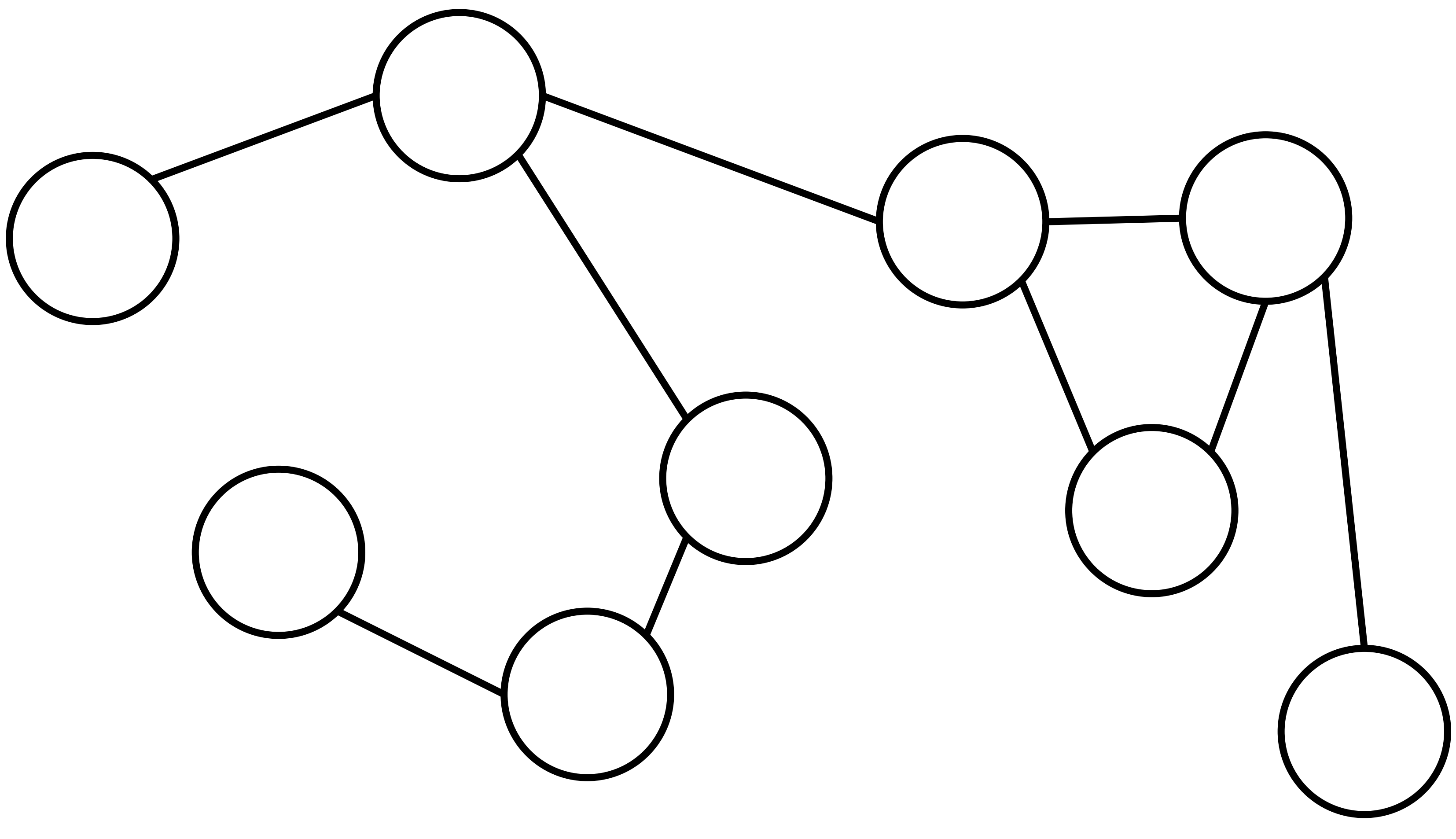
Case Study: Vertex Coloring



c-coloring:

a coloring of a graph with c (or less) colors

Case Study: Vertex Coloring



Chromatic number χ:

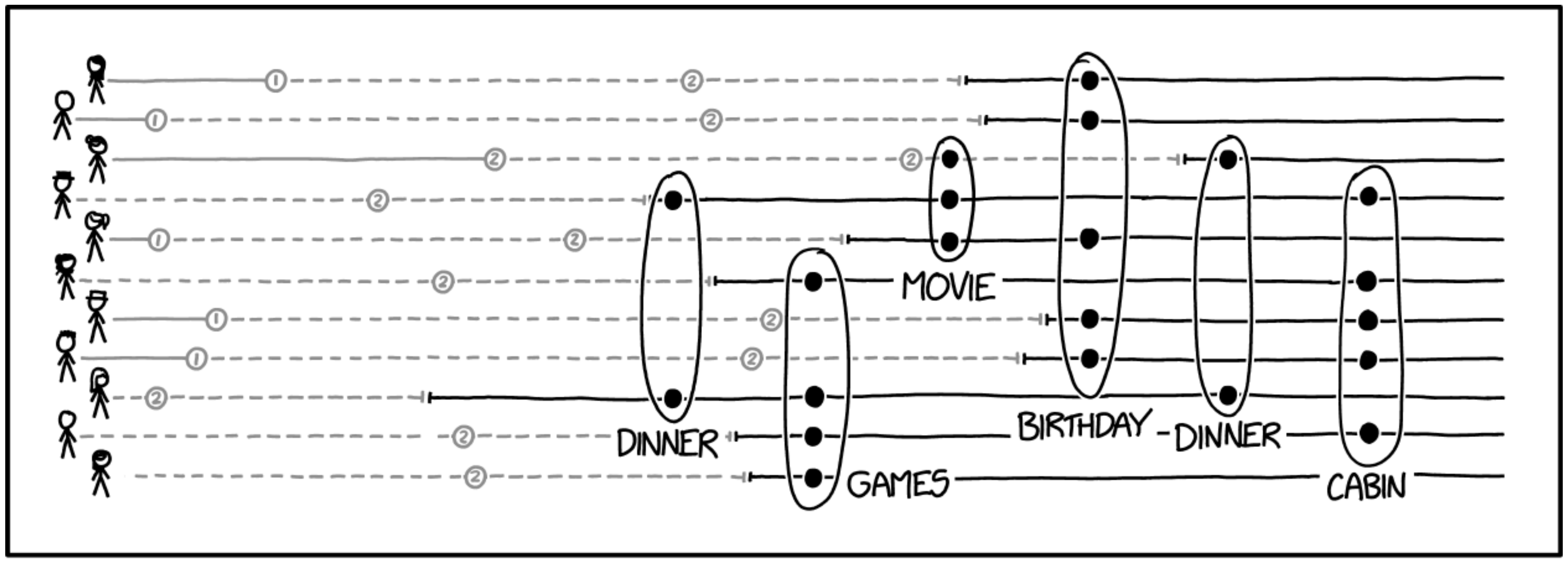
smallest number of colors needed to color a graph

Applications

Medium Access

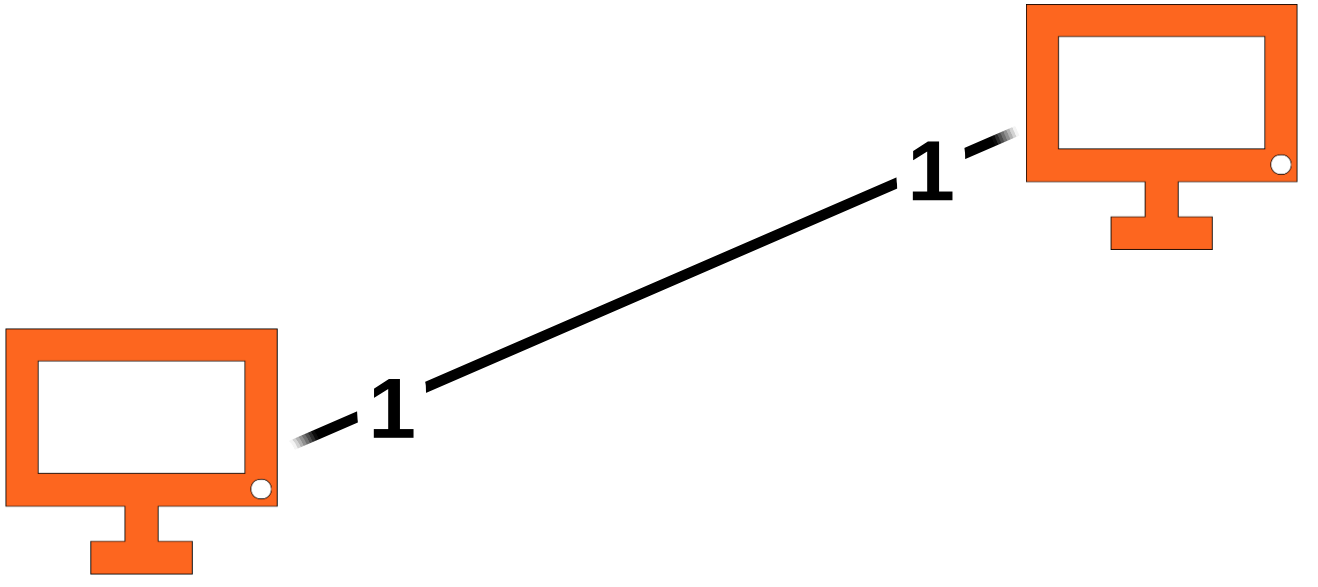
- Interference-free, efficient utilization of spectrum
- Neighboring cells should have different frequencies!
- Colors = frequencies, channels, etc.

Scheduling



POST-VACCINE SOCIAL SCHEDULING (xkcd #2450)

- As many jobs as possible should run in parallel
- Jobs that share a resource should be scheduled at a different time
- Colors = time slots, rounds
- Example: bipartite matching!

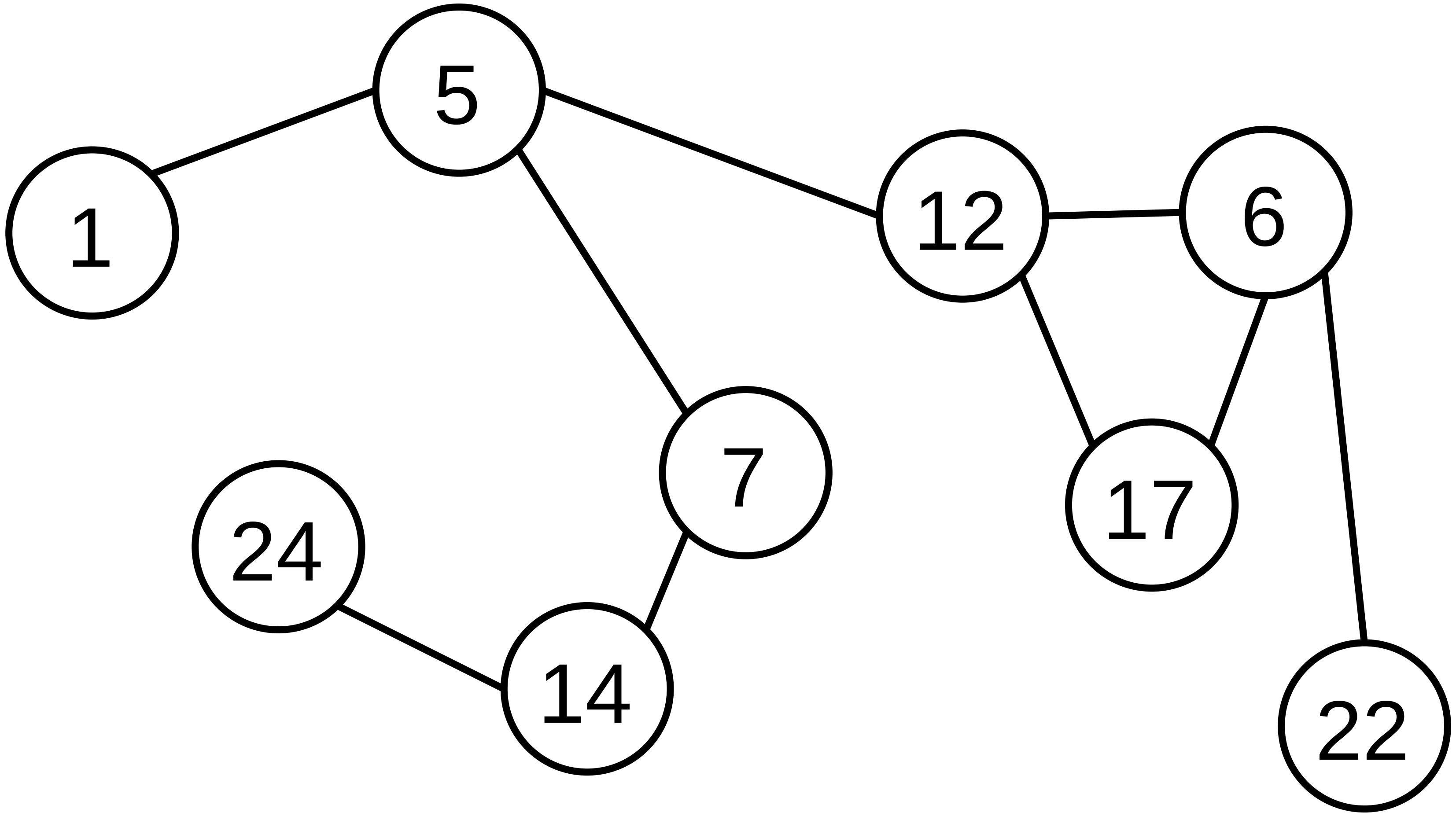


Last time: could not even color this graph in the port numbering model!

LOCAL model

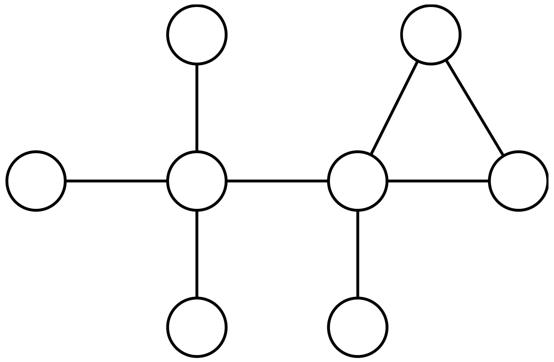
LOCAL model =
Port numbering model + unique IDs

LOCAL model

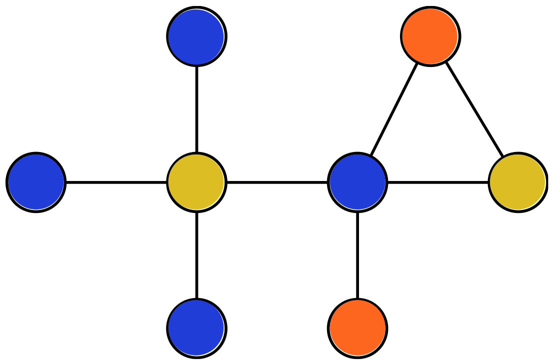


- Every node has a unique identifier
- In a system of n nodes, each identifier consists of $\log n$ bits (e.g. IP addresses)

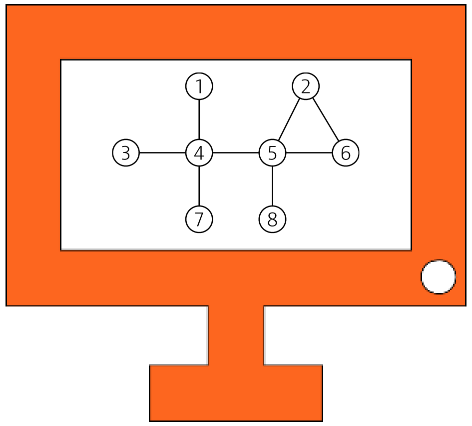
Centralized sequential coloring



Input: general graph

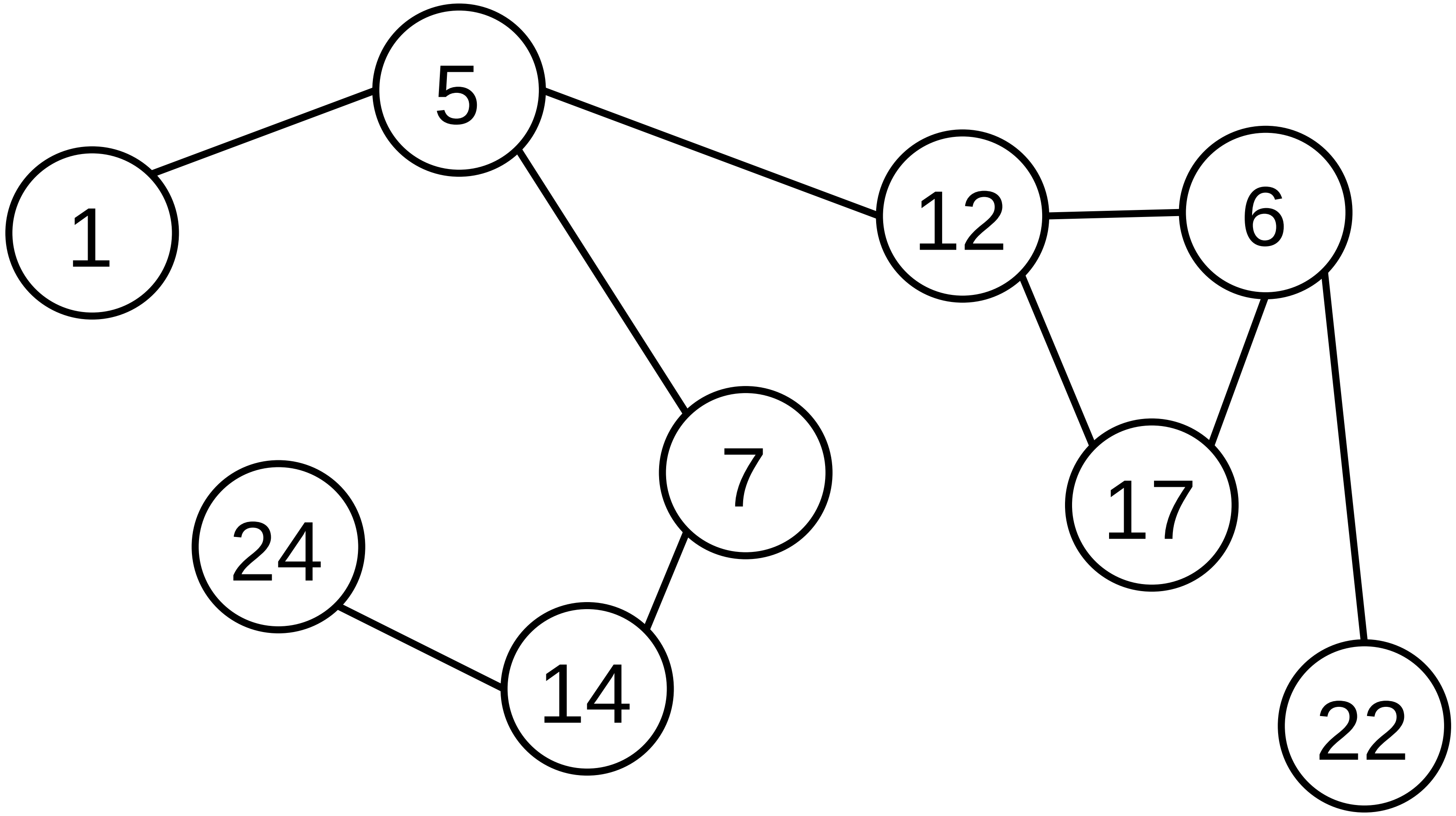


Output: $(\Delta + 1)$-coloring



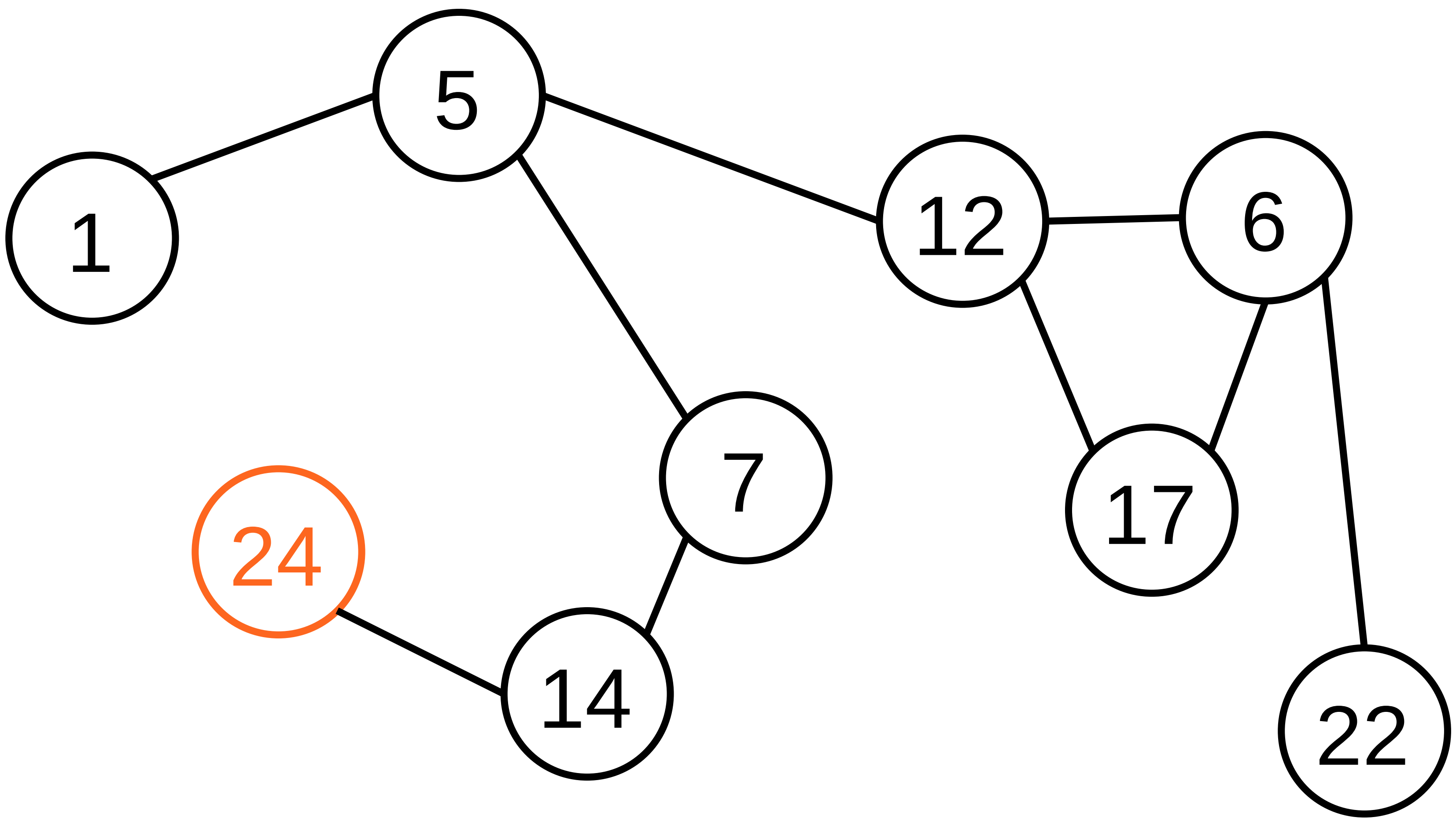
Model of computing:
Centralized & Node IDs

Centralized sequential coloring



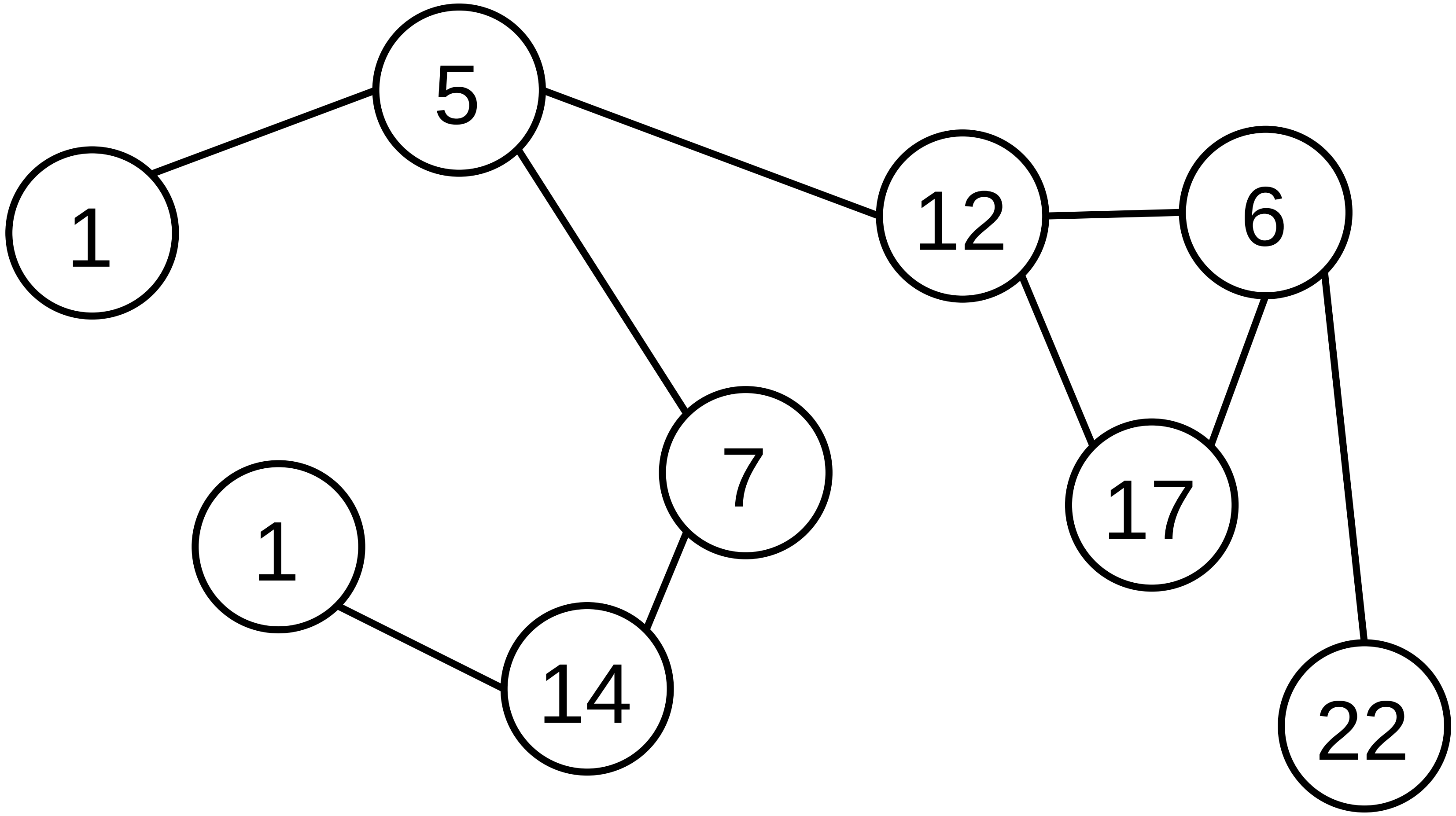
- Find node with the largest ID
- Recolor the node with the smallest color in $\{1, \ldots, \Delta + 1\}$ that does not cause a conflict

Centralized sequential coloring

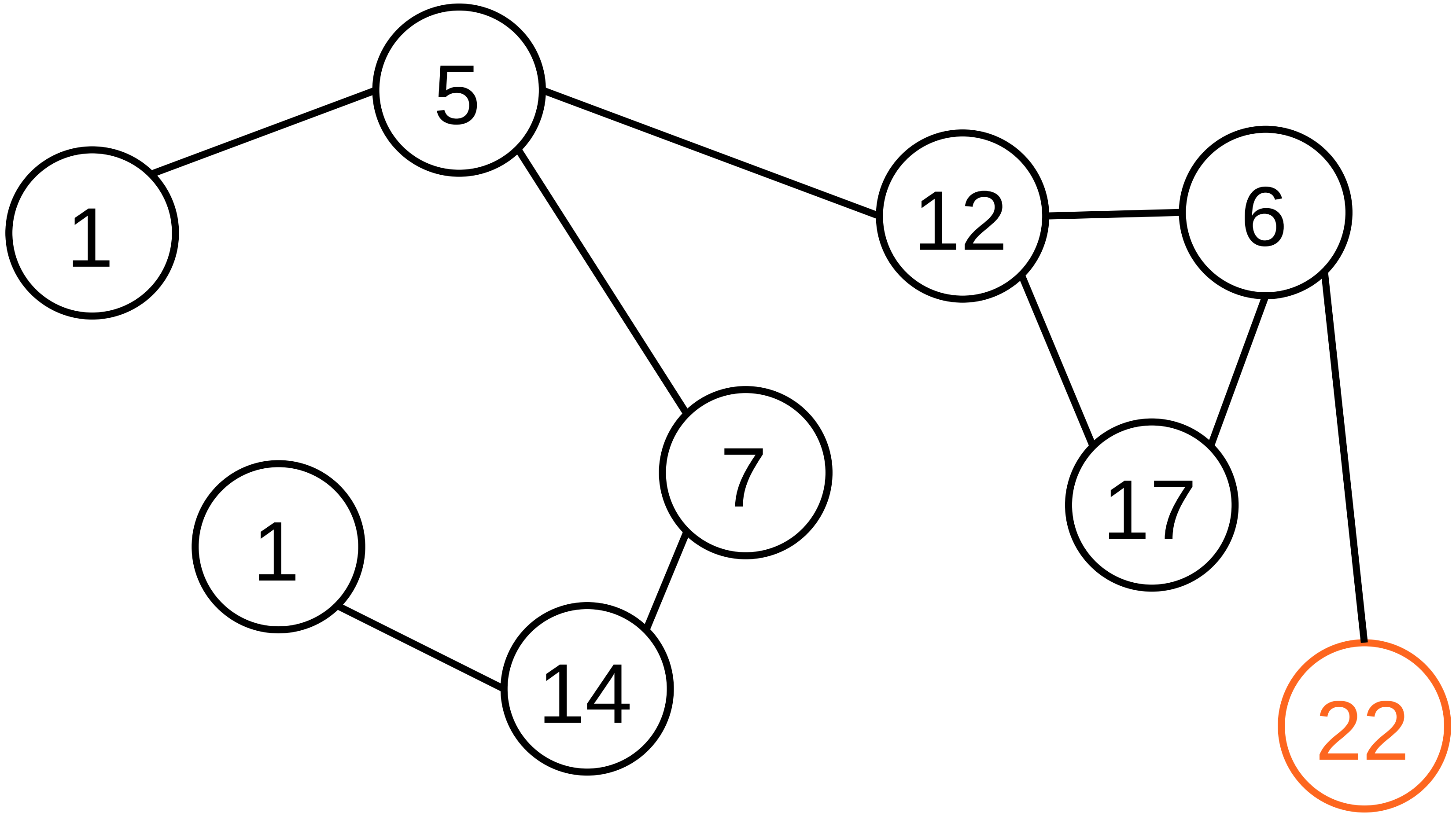


- Find node with the largest ID
- Recolor the node with the smallest color in $\{1, \ldots, \Delta + 1\}$ that does not cause a conflict

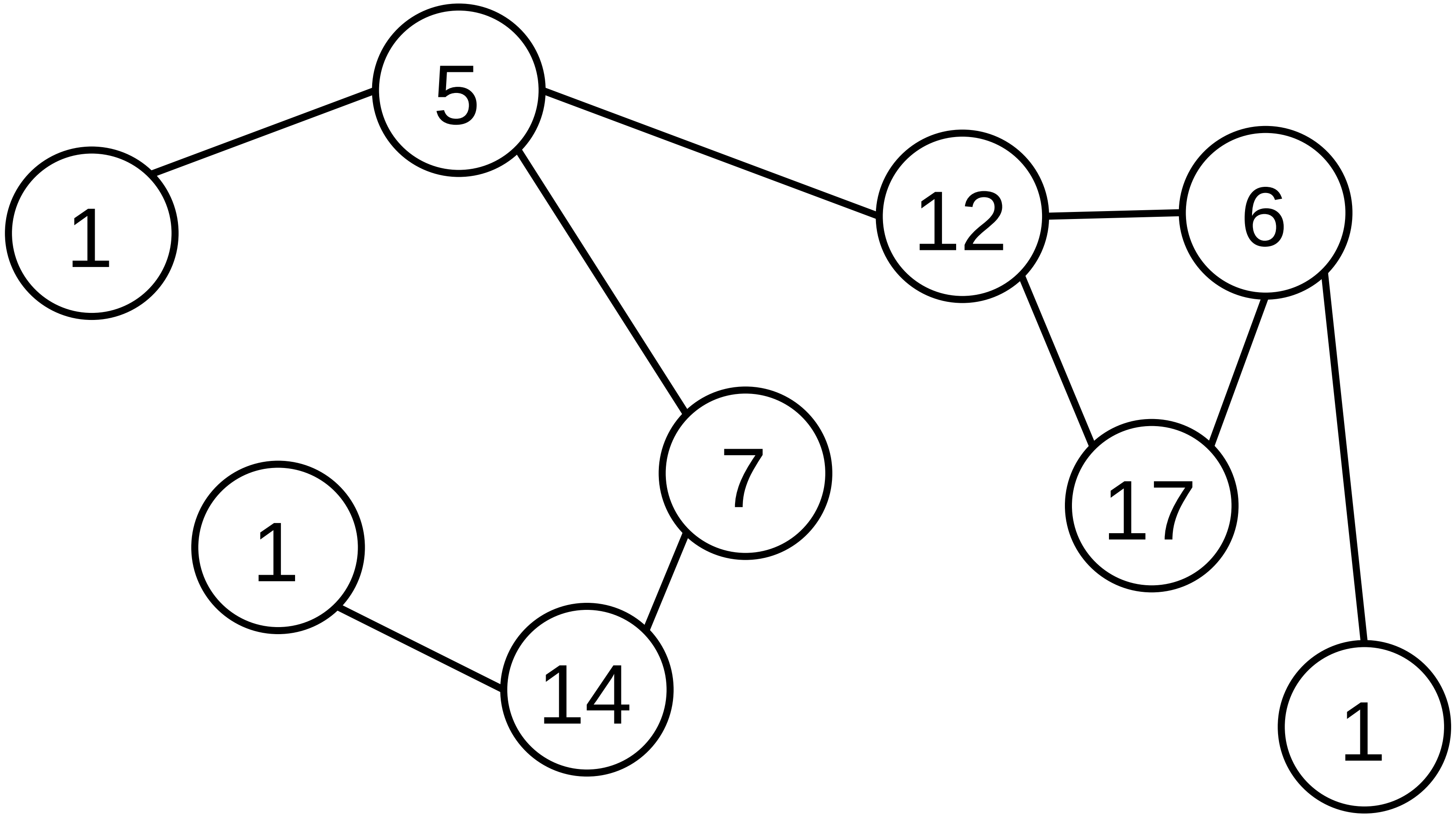
Centralized sequential coloring



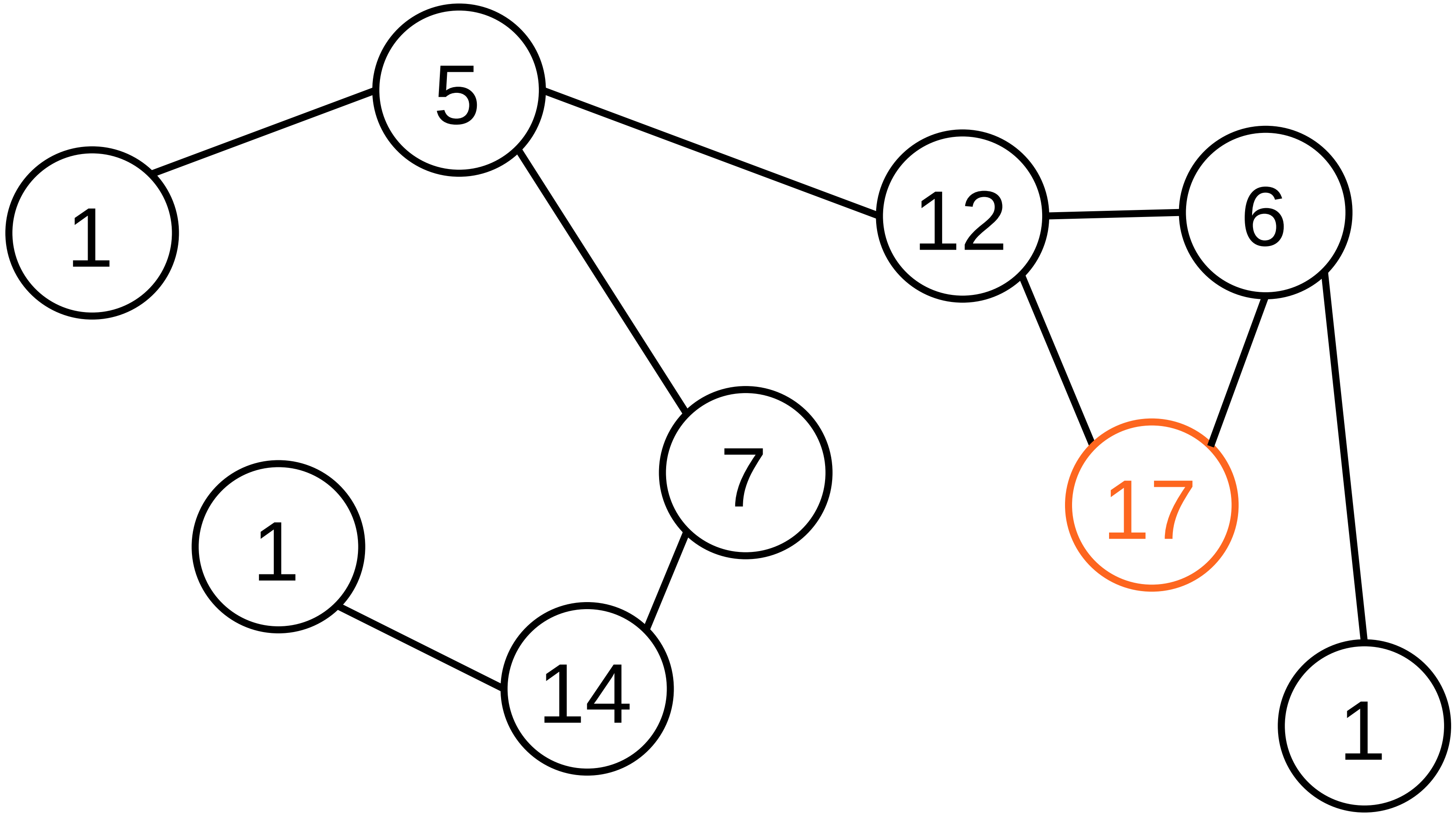
Centralized sequential coloring



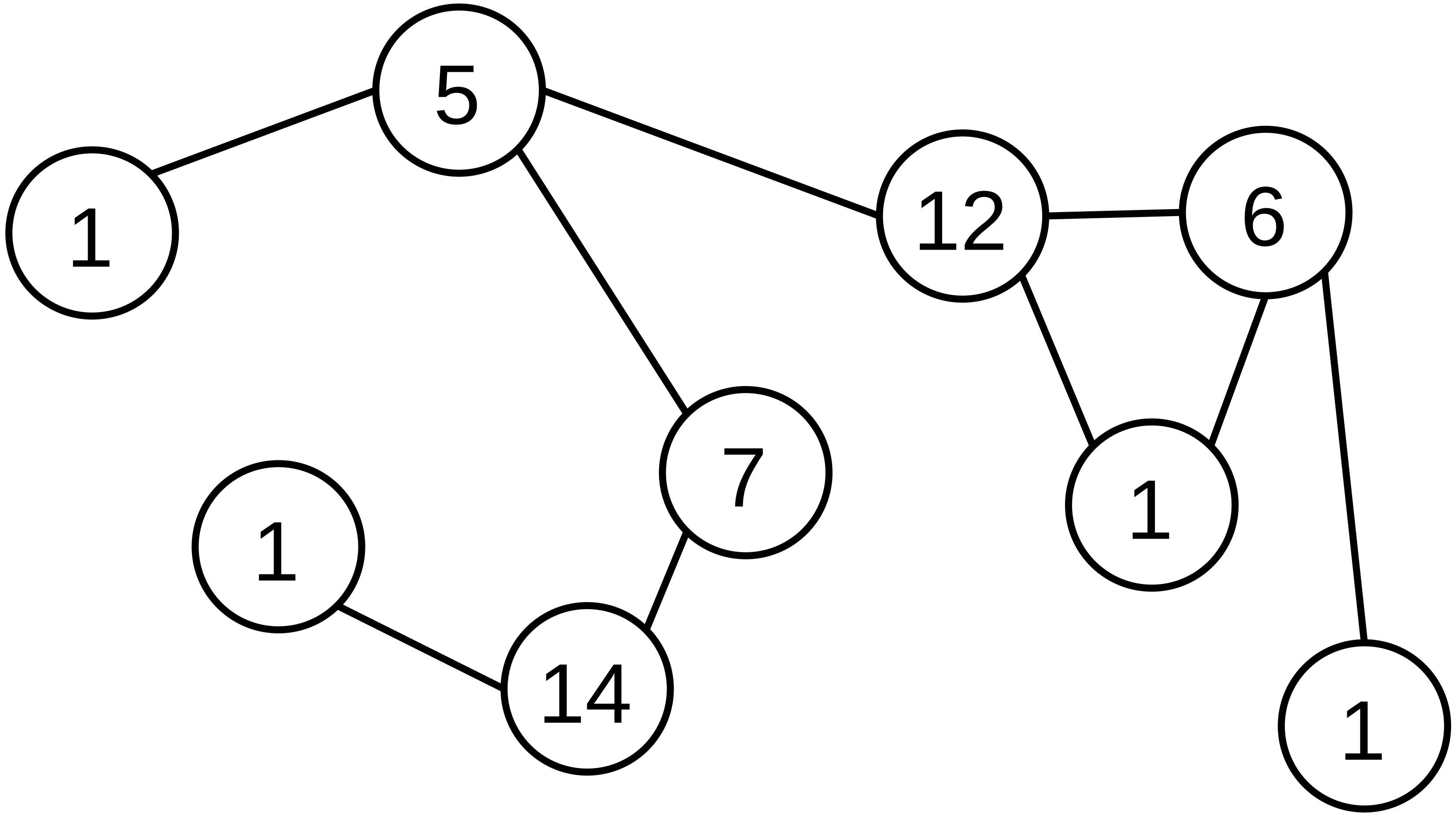
Centralized sequential coloring



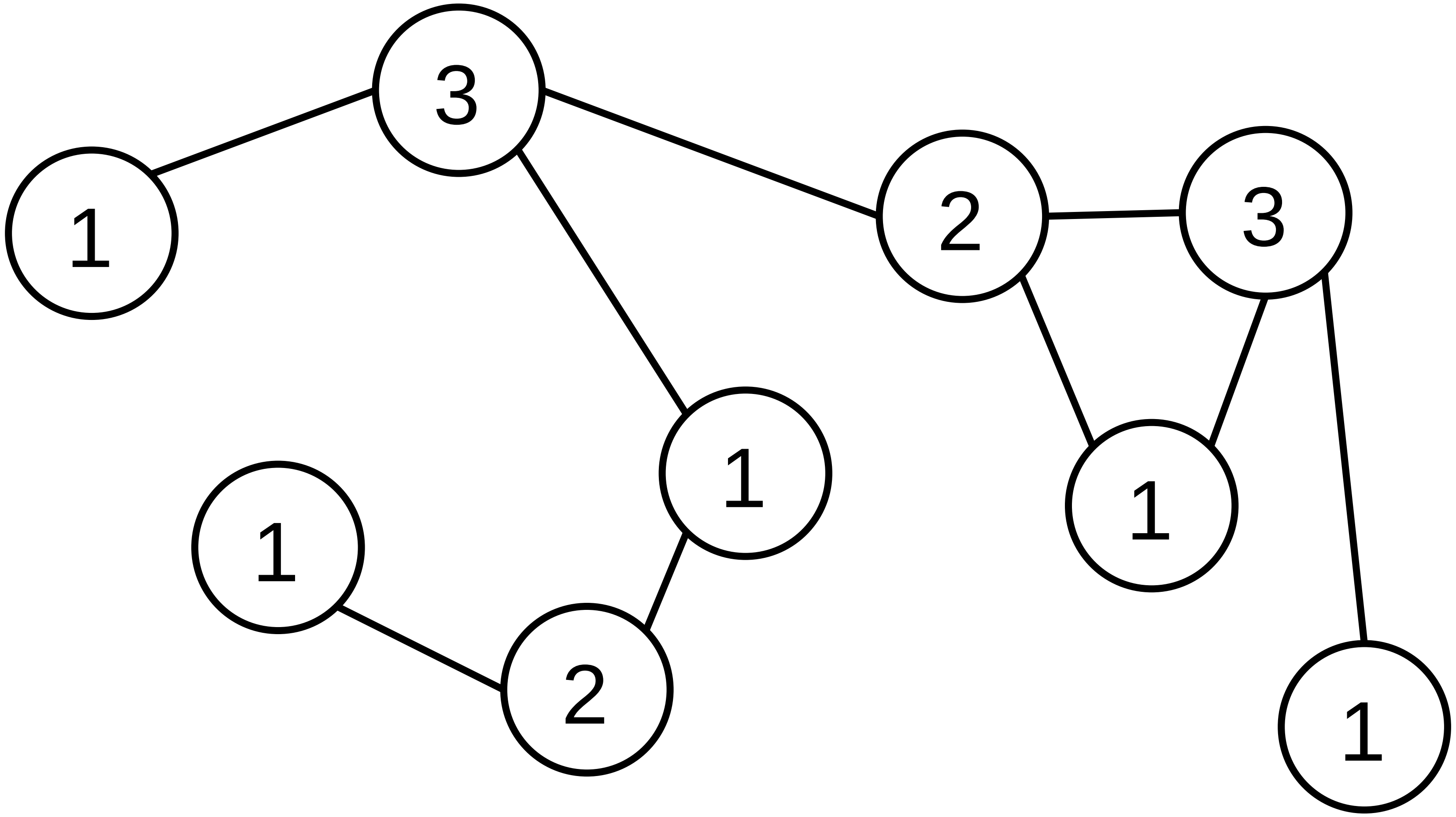
Centralized sequential coloring



Centralized sequential coloring

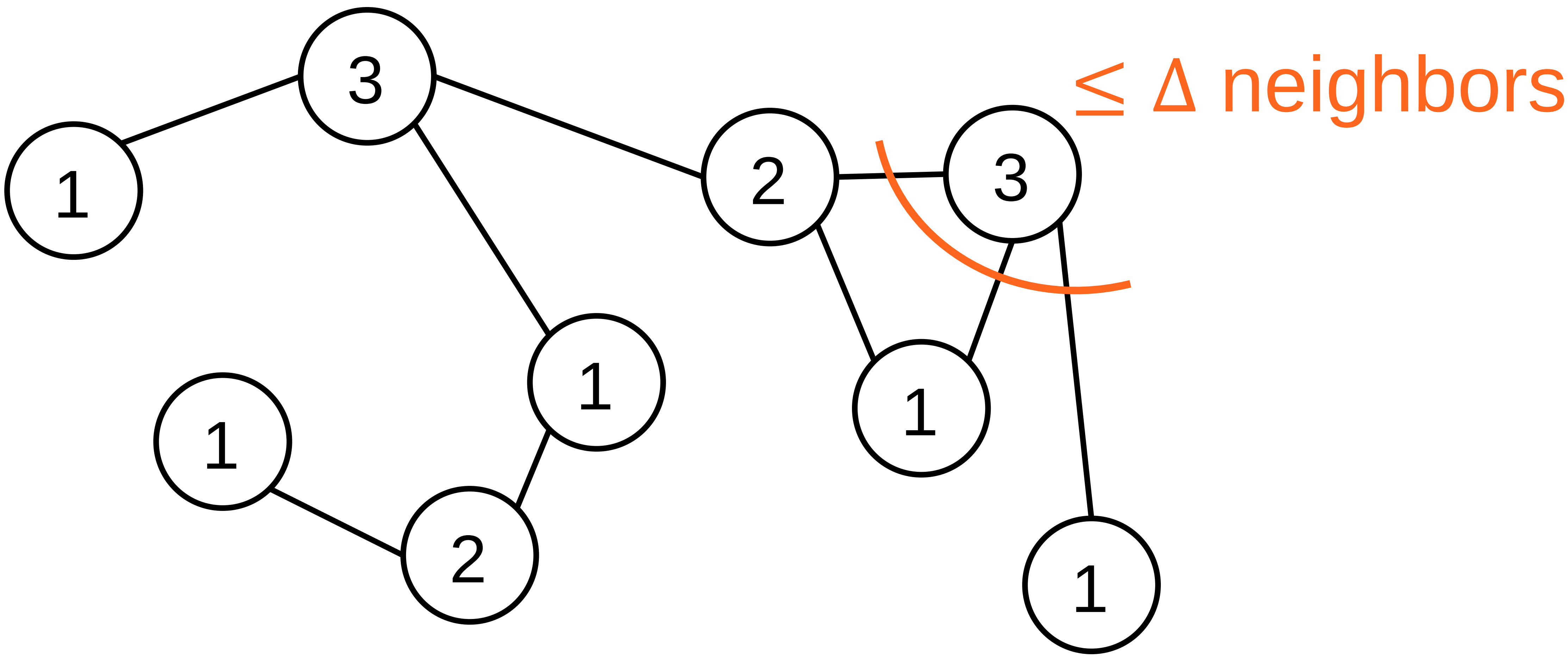


Centralized sequential coloring



How many colors do we get?

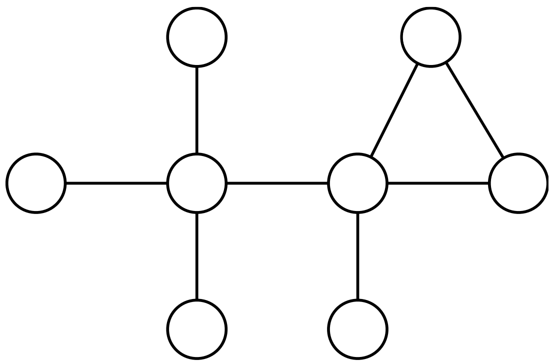
Centralized sequential coloring



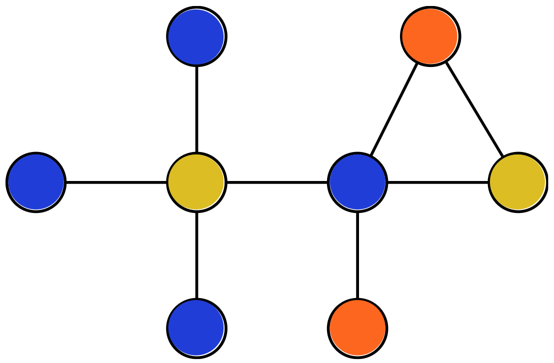
How many colors do we get?

Every node has at most Δ neighbors, so at most $\Delta + 1$ colors

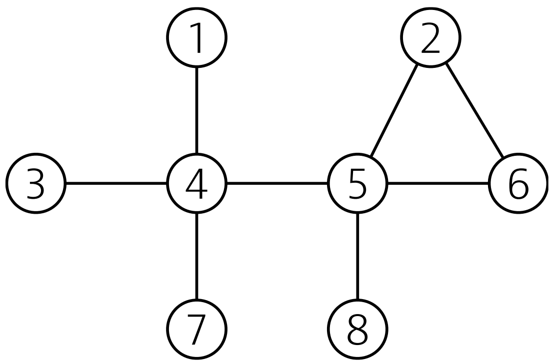
Distributed coloring: Reduce



Input: general graph

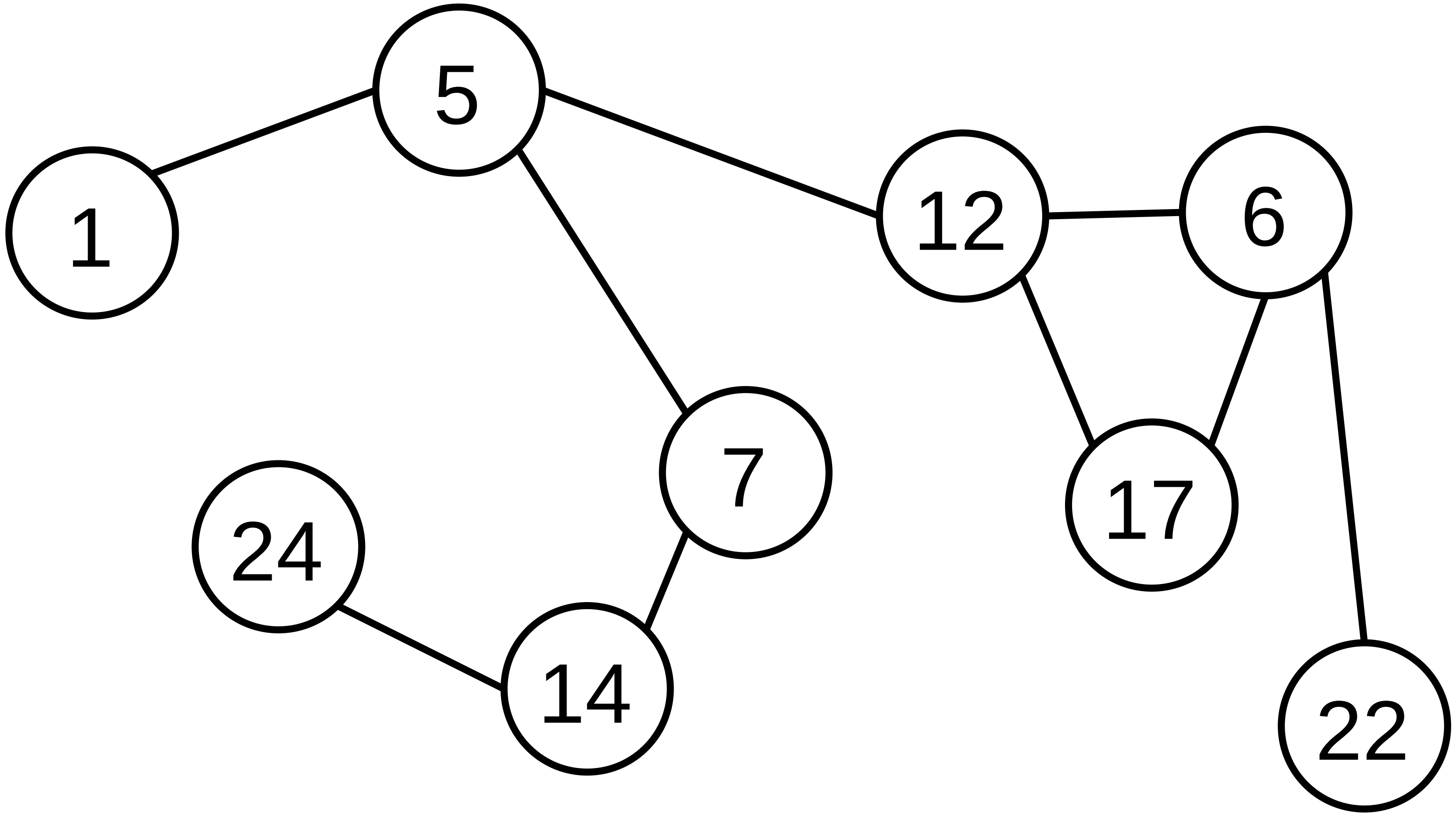


Output: $(\Delta + 1)$-coloring



Model of computing:
synchronous LOCAL model

Reduce



- Send color to neighbors
- If all neighbors have a smaller color:
  Recolor the node with the smallest color in $\{1, \ldots, \Delta + 1\}$ that does not cause a conflict

Performance Metrics for Distributed Algorithms

Time Complexity:

Number of communication rounds

Message Complexity:

Number of messages sent

Local Computation:

Complexity of local computations

Quality of solution:

Approximation ratio

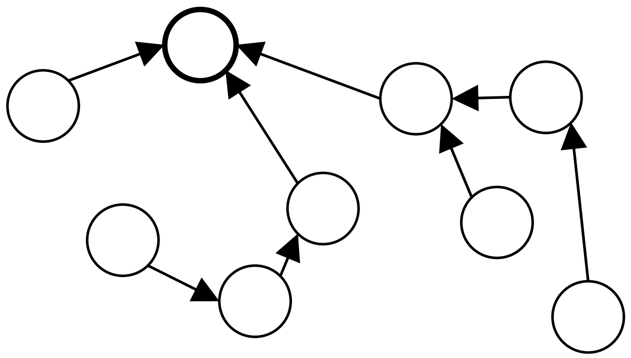
Analysis: Reduce

Simpler: Coloring rooted trees

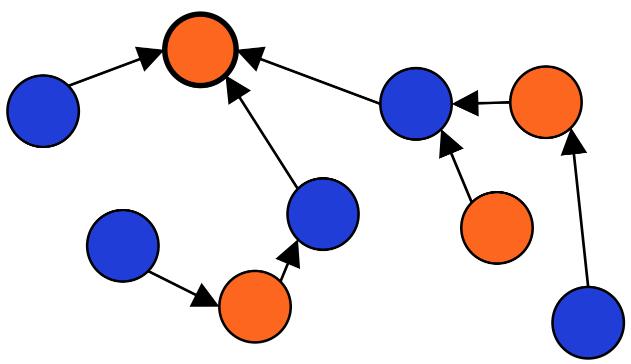
Tree =
Acyclic connected graph (with $\chi \leq 2$)

Rooted tree =
Tree where one node is designated the root; every edge is directed toward the root

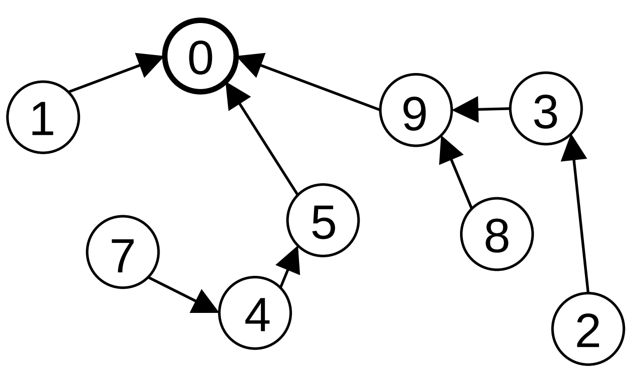
Slow tree coloring



Input: rooted tree

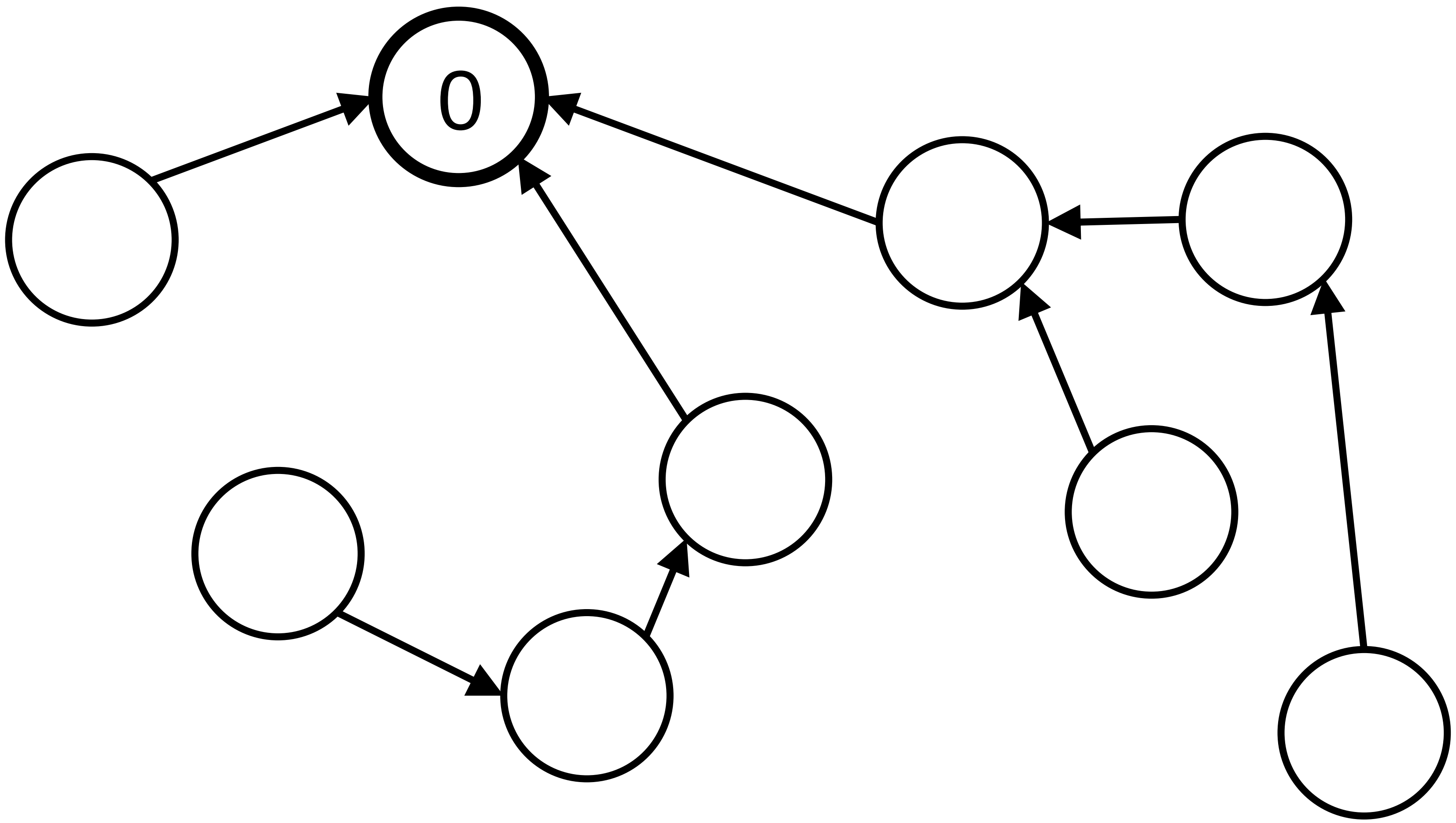


Output: 2-coloring



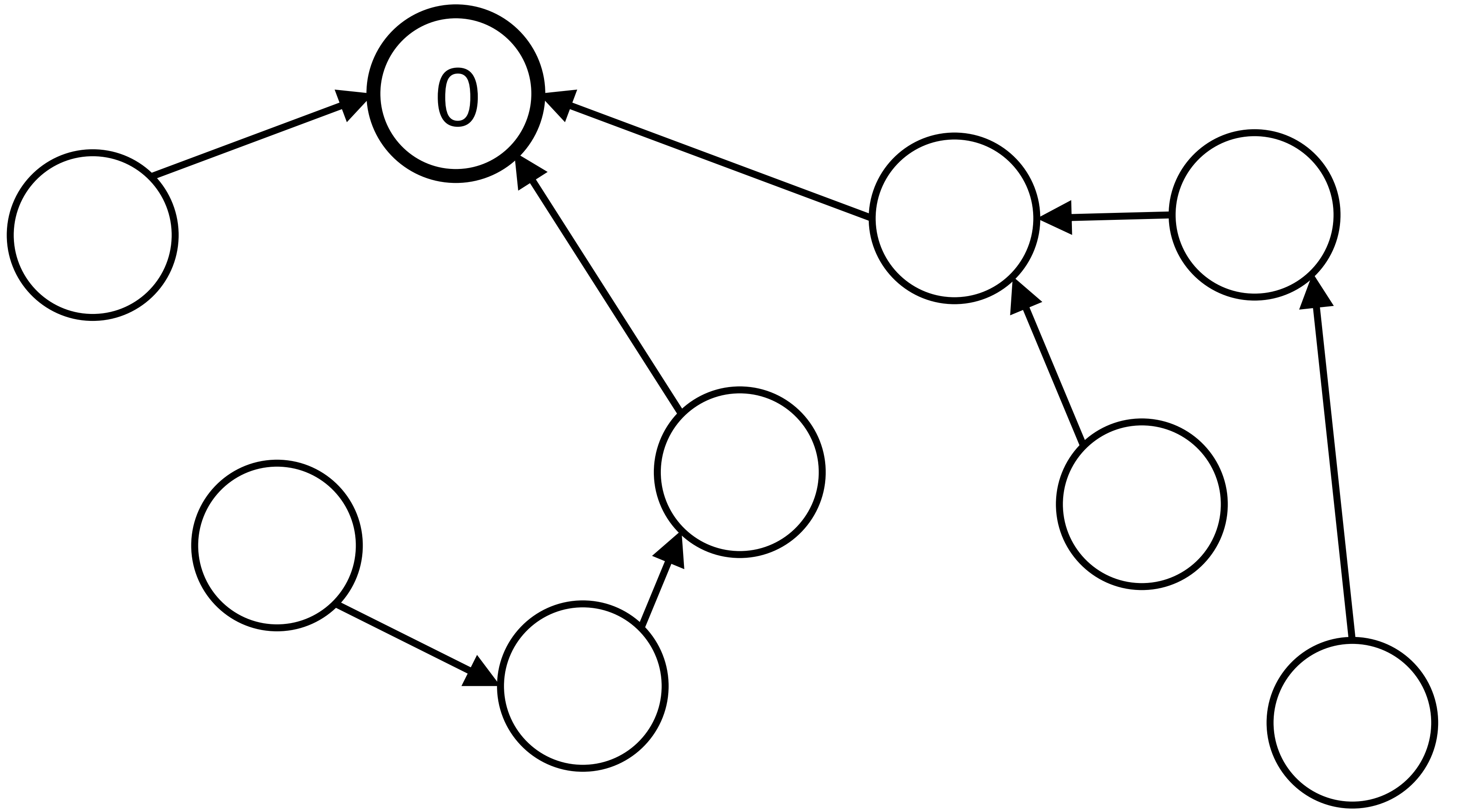
Model of computing:
synchronous LOCAL model

Slow tree coloring



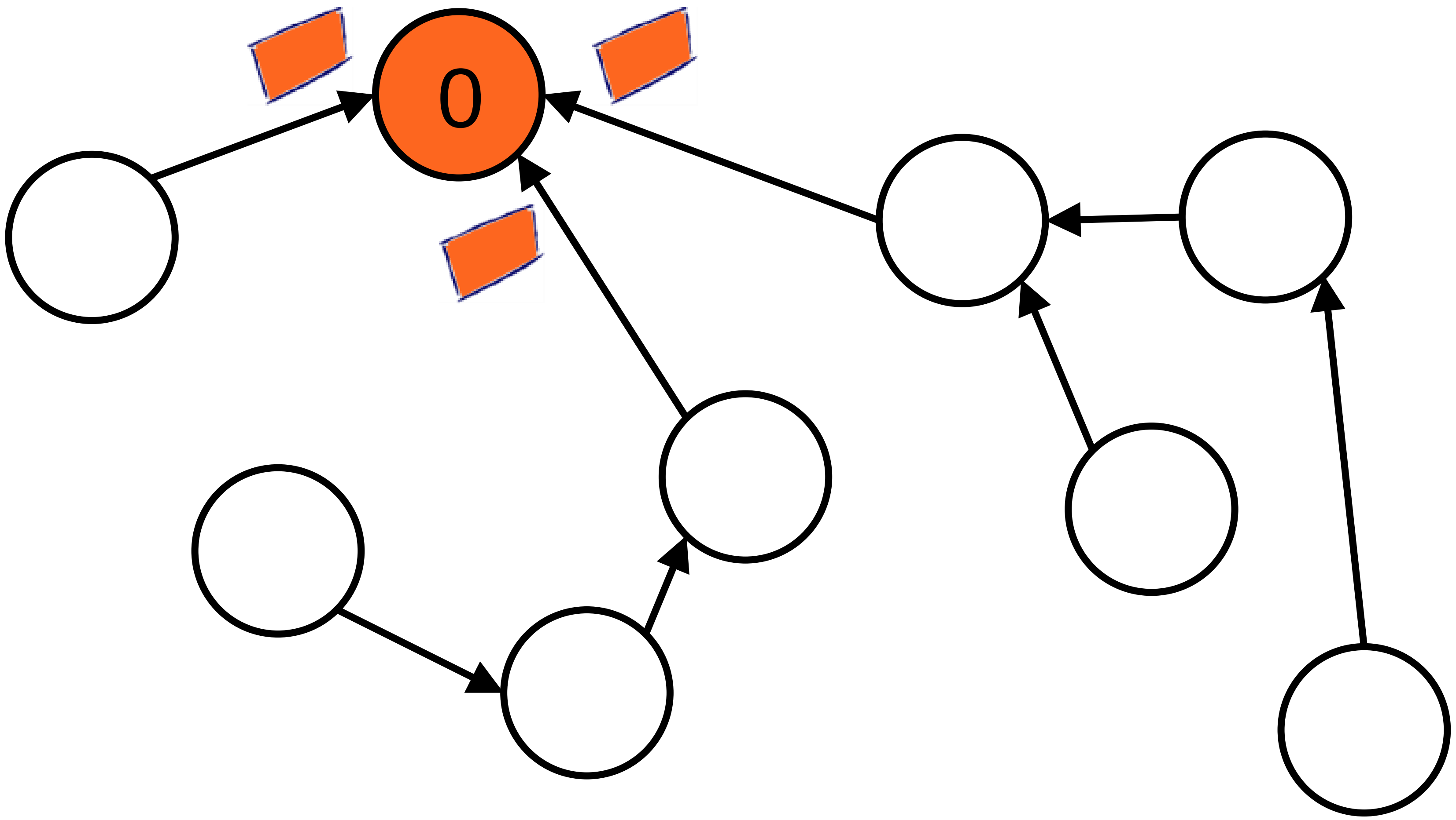
- Rooted tree with Root ID 0

Slow tree coloring



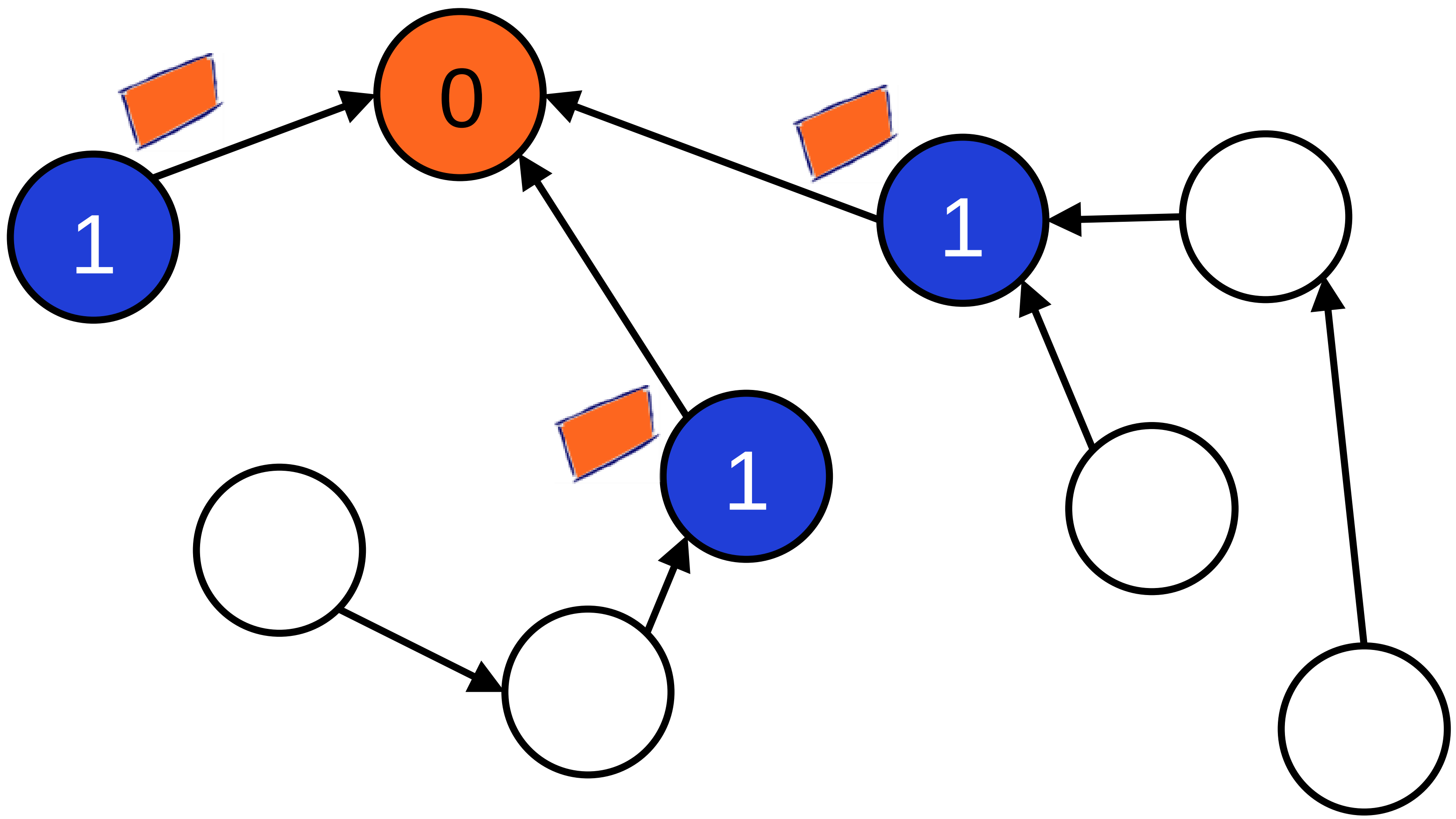
- Rooted tree with Root ID 0
- Idea: interpret root bit as color! Iteratively communicate colors to children and take opposite color from parent!

Slow tree coloring



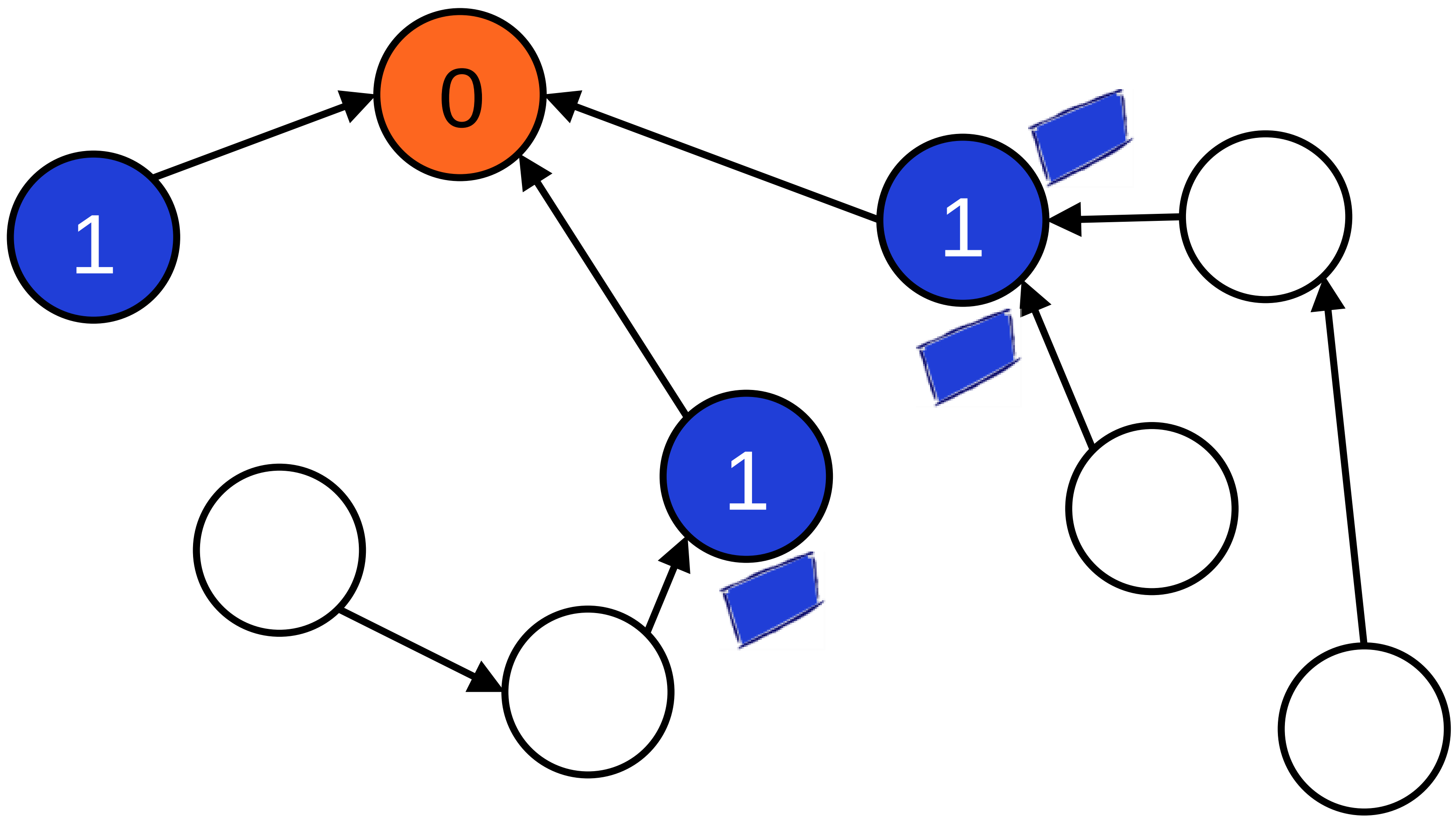
Round 1

Slow tree coloring



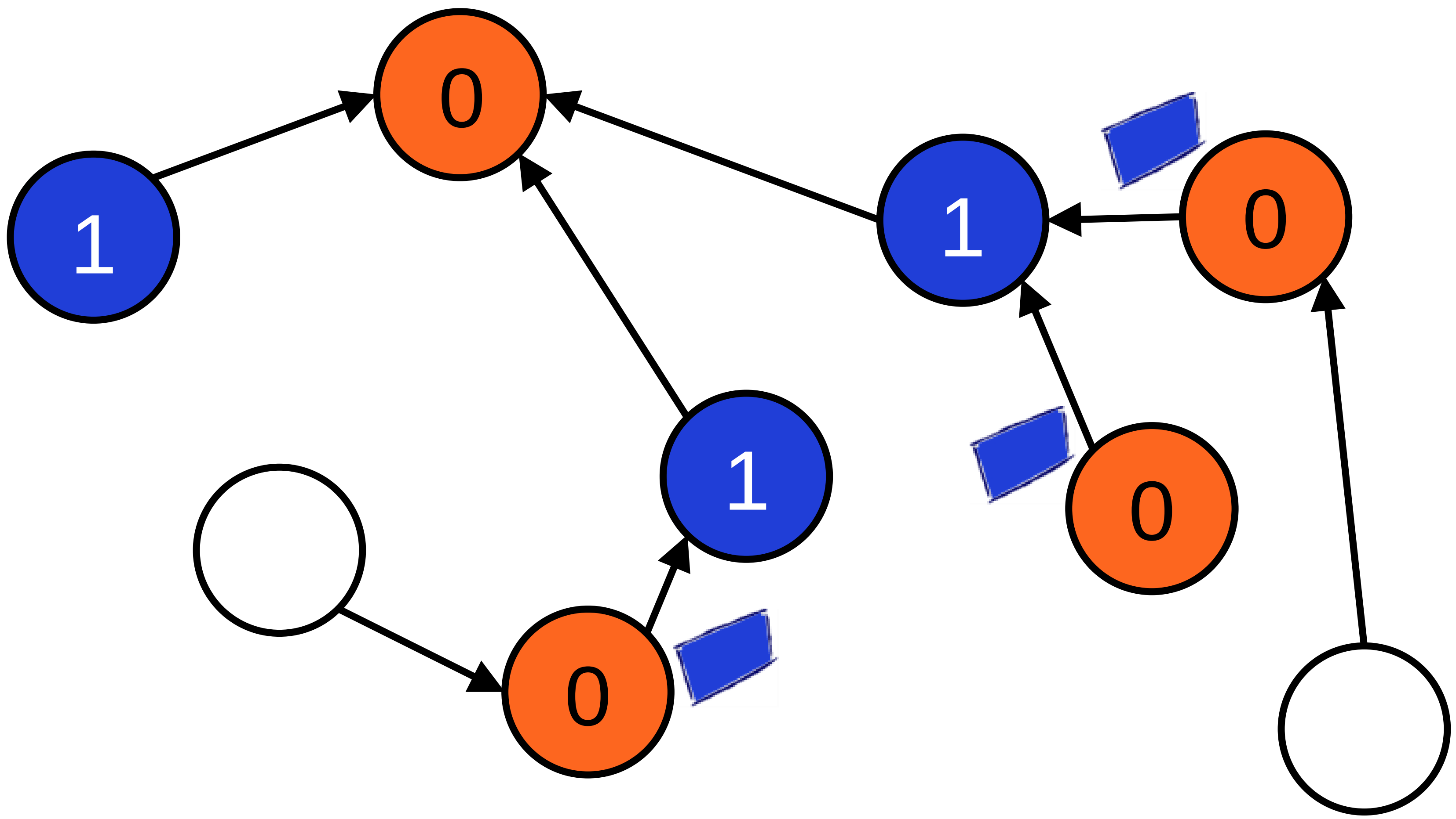
Round 1

Slow tree coloring



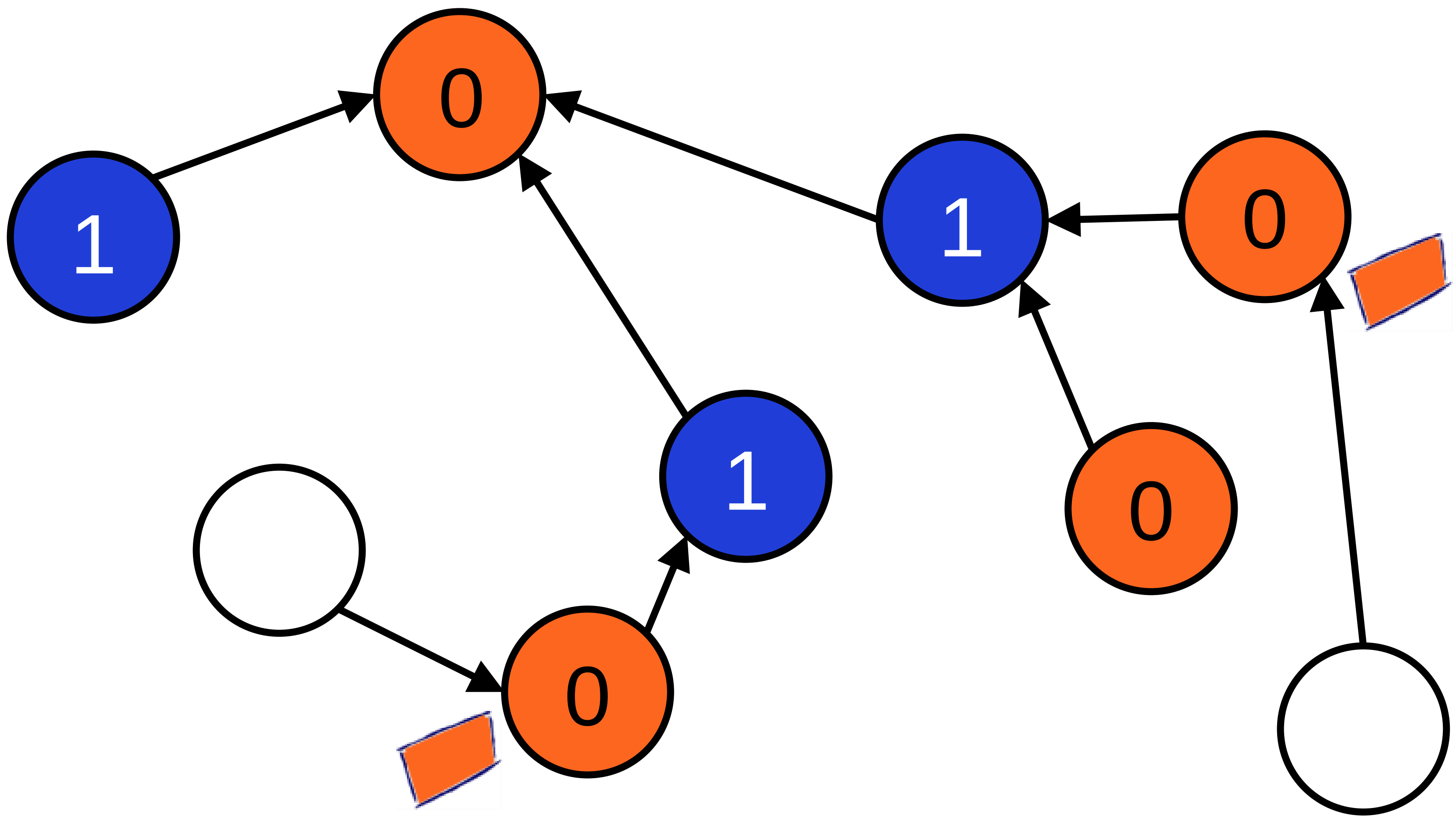
Round 2

Slow tree coloring



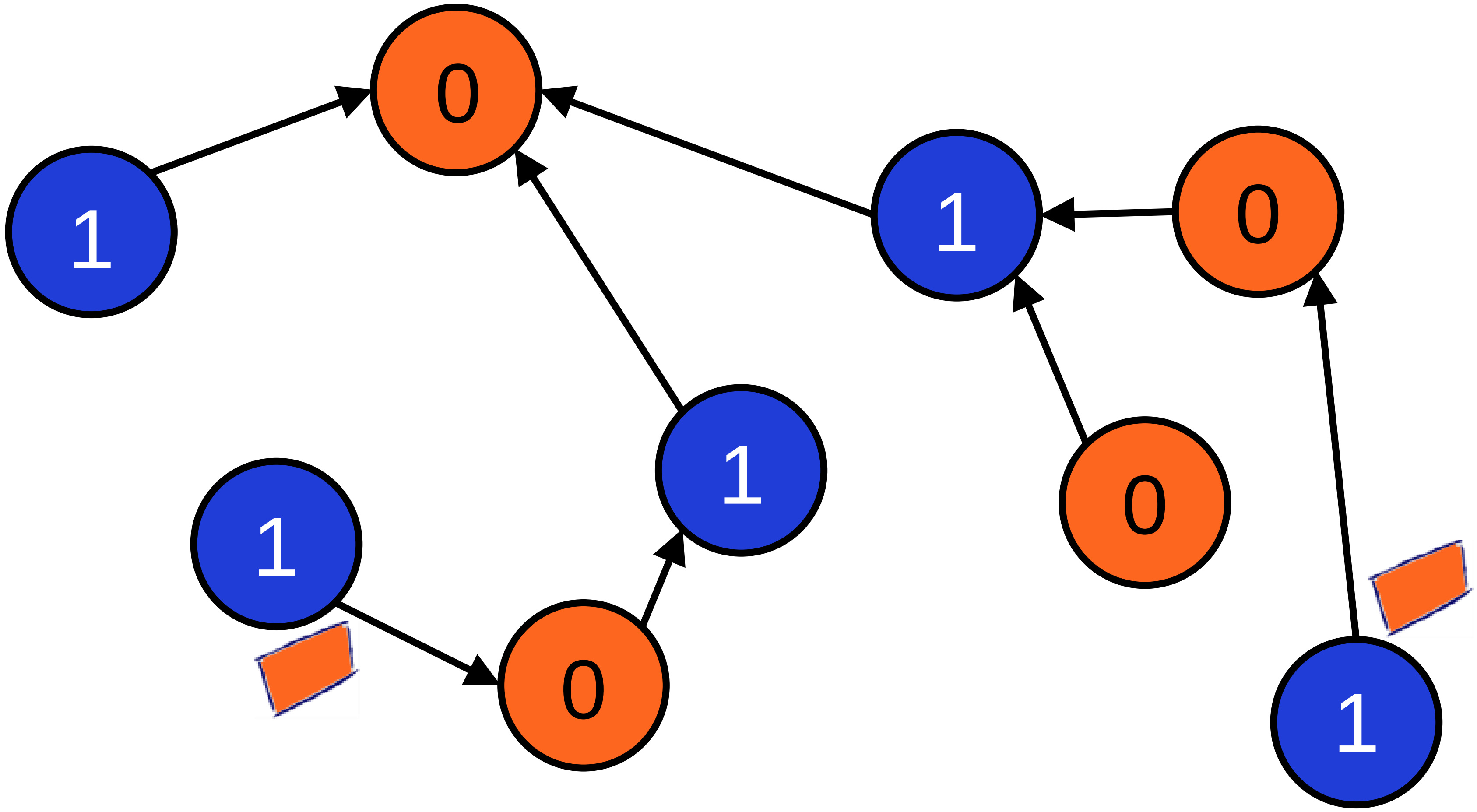
Round 2

Slow tree coloring



Round 3

Slow tree coloring



Round 3

Slow Tree Coloring

If root: color 0, send 0 to children

Otherwise: each node v:

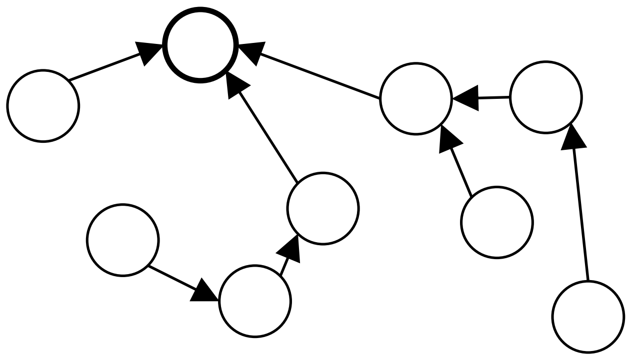
  Wait for message x from parent

  Choose color $y = 1 - x$

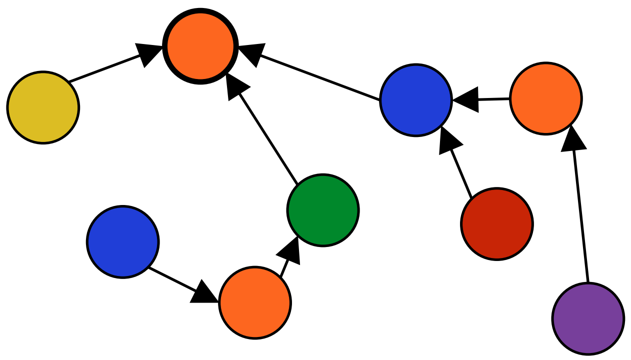
  Send y to children

Analysis: Slow tree coloring

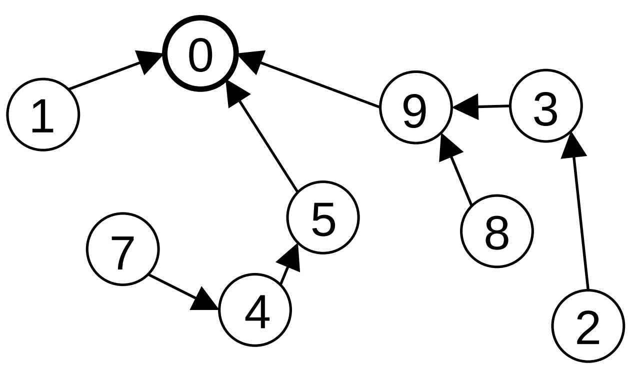
Fast tree coloring



Input: rooted tree



Output: 6-coloring



Model of computing:
synchronous LOCAL model

From 2^x to $2x$ colors

Assume: each node has an ID in $\{1,\ldots,2^x\}$

Each node v does (in parallel):

send own color c_v to the children

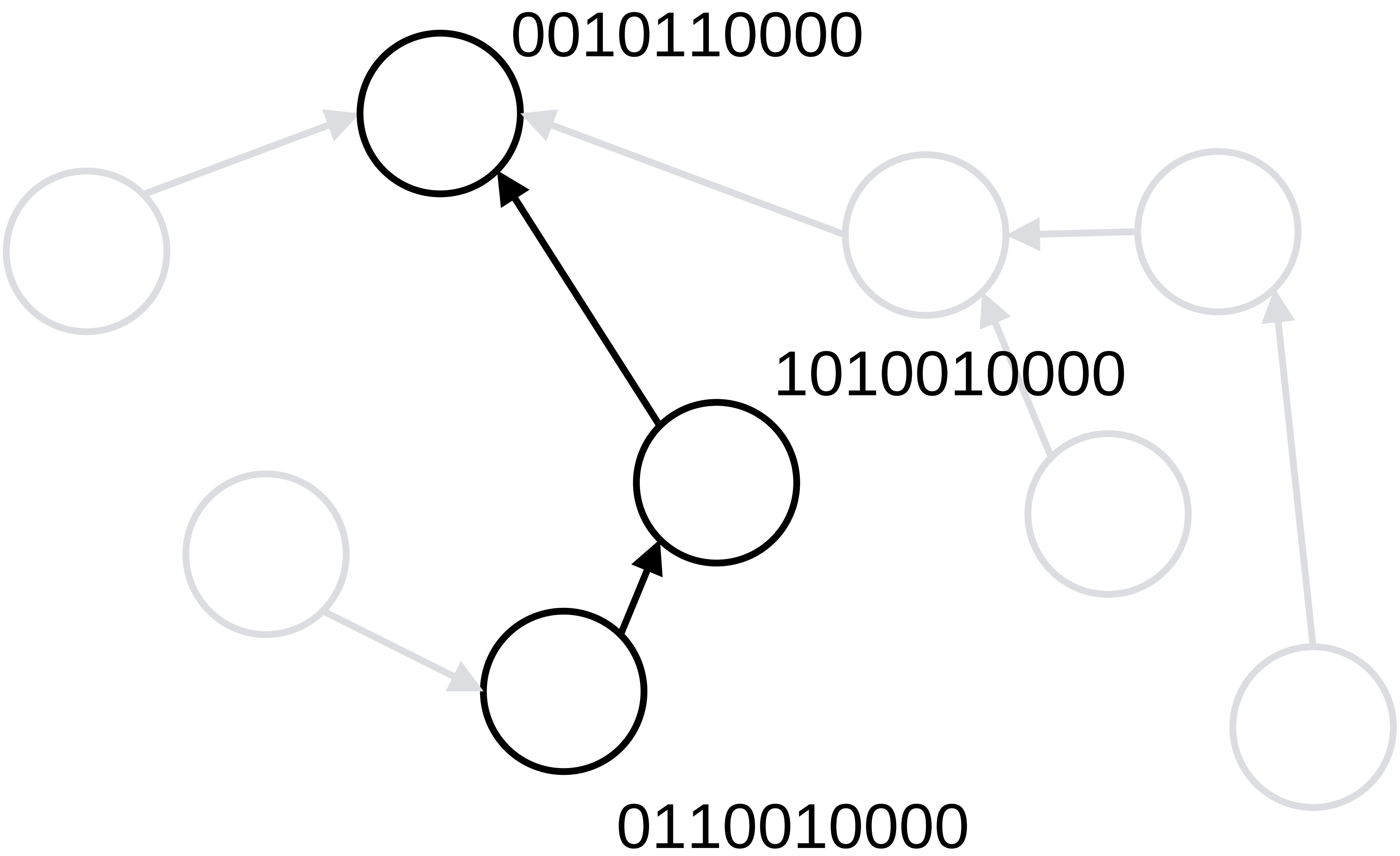
receive color c_p of the parent

let i be the index of the first bit where c_v and c_p differ

let b be the value of the bit of c_v that differs

set new color to $c_v = 2i + b$

Example



Analysis: Color reduction from 2^x to $2x$

6-Color

Assume: each node has an ID in $\{1, \ldots, n\}$

repeat

apply color reduction "from 2^x to $2x$" colors

until $c_v \in \{0, \ldots, 5\}$ for all nodes

6-Color

Assume: each node has an ID in $\{1, \ldots, n\}$

repeat

apply color reduction "from 2^x to $2x$" colors

until $c_v \in \{0, \ldots, 5\}$ for all nodes

Logarithmic reduction of colors in every round!

Time complexity: $O(\log^* n)$

Intuition: n vs. $O(\log^* n)$

$\log n$: How many times do I have to $/2$ until < 2?

$$n, n/2, n/4, n/8, \ldots, 8, 4, 2, 1$$

$$\xleftrightarrow{\hspace{6em}}$$

$$\log n$$

Intuition: n vs. $O(\log^* n)$

$\log n$: How many times do I have to $/2$ until < 2?

$$\underbrace{n, n/2, n/4, n/8, \dots, 8, 4, 2, 1}_{\log n}$$

$\log \log n$: How many times do I have to $\sqrt{x}$ until < 2?

$$\underbrace{n, \sqrt{n}, \sqrt{\sqrt{n}}, \sqrt{\sqrt{\sqrt{n}}}, \dots}_{\log \log n}$$

Intuition: n vs. $O(\log^* n)$

$\log n$: How many times do I have to $/2$ until < 2?

$$\underbrace{n, n/2, n/4, n/8, \dots, 8, 4, 2, 1}_{\log n}$$

$\log \log n$: How many times do I have to $\sqrt{x}$ until < 2?

$$\underbrace{n, \sqrt{n}, \sqrt{\sqrt{n}}, \sqrt{\sqrt{\sqrt{n}}}, \dots}_{\log \log n}$$

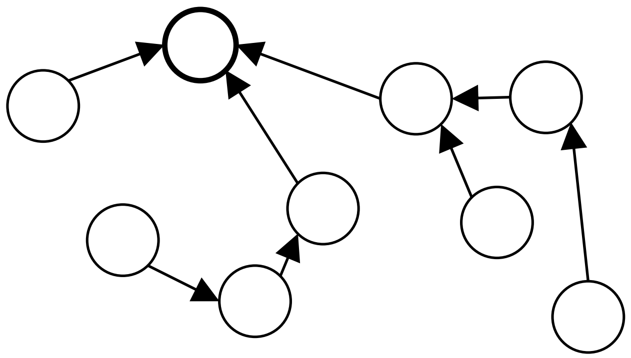
$\log^* n$: How many times do I have to $\log x$ until < 2?

$$\underbrace{n, \log n, \log \log n, \log \log \log n, \dots}_{\log^* n}$$

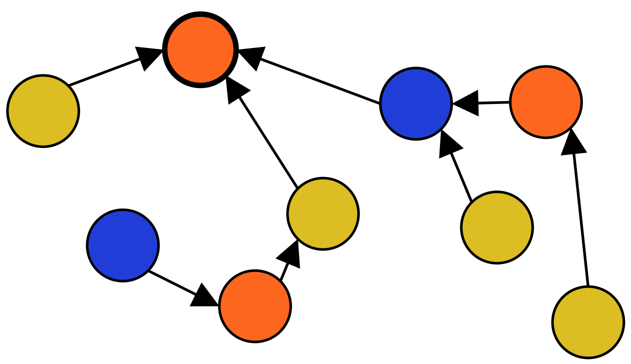
n = atoms in the universe $\approx 10^{80}$

$\log^*$(atoms in the universe) ≈ 5

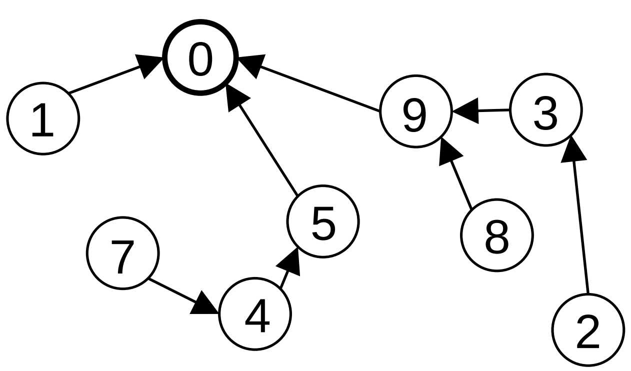
Fast 3-coloring in trees



Input: rooted tree



Output: 3-coloring



Model of computing:
synchronous LOCAL model

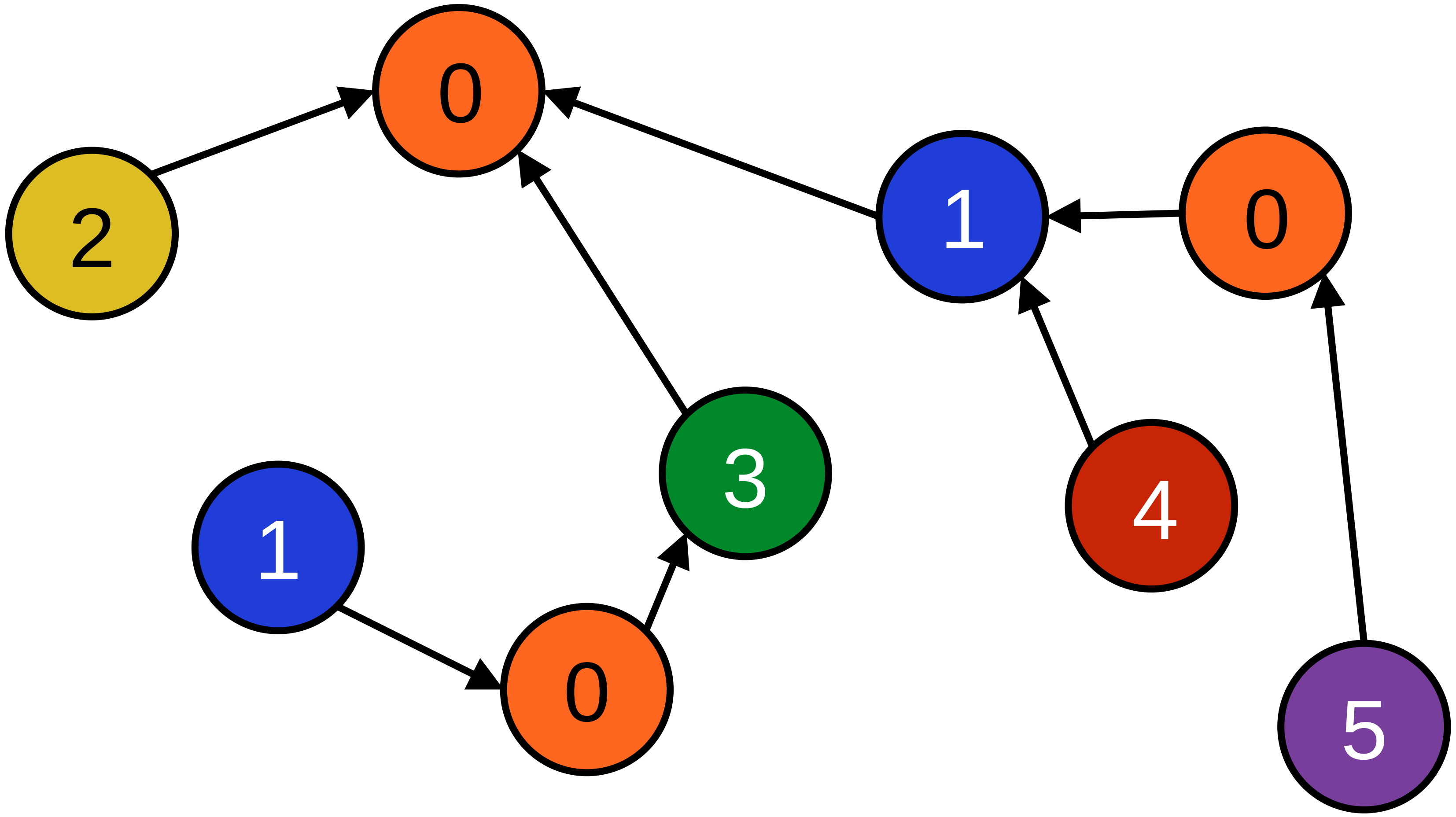
Subroutine: Shift down

Each node v (not root) concurrently does:

recolor v with color of parent

The root chooses the smallest free color

Example



Six-2-Three

Each other node v does (in parallel):

Run „**6-Colors**“ for $\log^*(n)$ rounds

For $x = 5,4,3$:

Perform „**Shift Down**“

If $(c_v = x)$ choose new color $c_v \in \{0,1,2\}$ according to „**Reduce**“

Analysis: 3-coloring trees

Learning goals

- **Graph problems:** coloring
- **Distributed models:** synchronous LOCAL model
- **Algorithms:**
  - $(\Delta + 1)$-coloring any graph (Reduce)
  - 2-coloring rooted trees (Slow Tree Coloring)
  - 6-coloring rooted trees (6-Coloring)
  - Shift Down technique on trees
  - 3-coloring rooted trees (Six-2-Three)