# MPRI Course 2.18.1
# **Distributed algorithms on networks**

**Pierre Fraigniaud**

INSTITUT
DE RECHERCHE
EN INFORMATIQUE
FONDAMENTALE

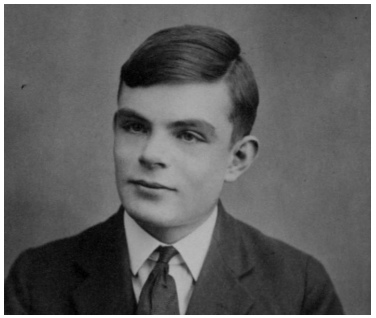
https://www.irif.fr/~pierref

MPRI, Université Paris Cité, 2024-2025

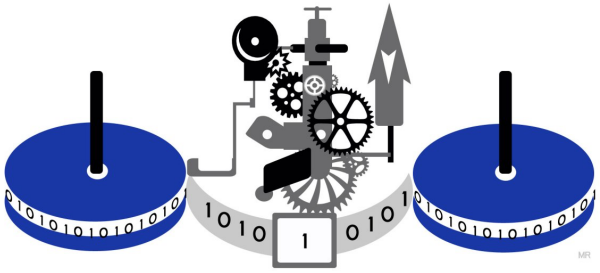
# Distributed Algorithms

A collection of autonomous computing entities
collaborating for solving a task
in absence of any coordinator

# Sequential vs. Distributed

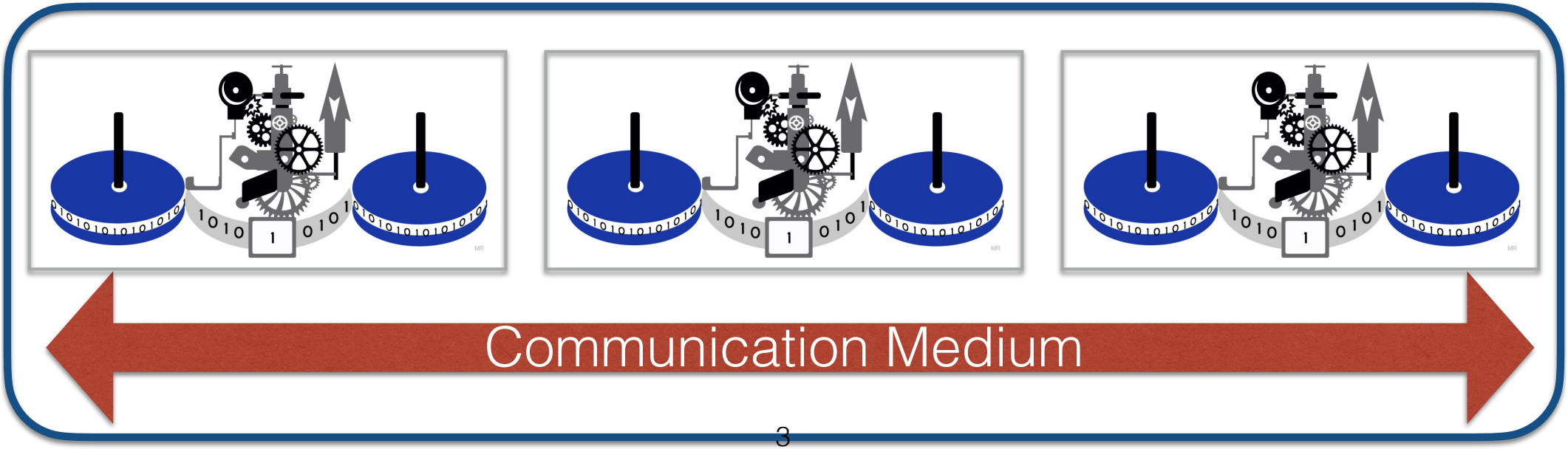


**Alan Turing**

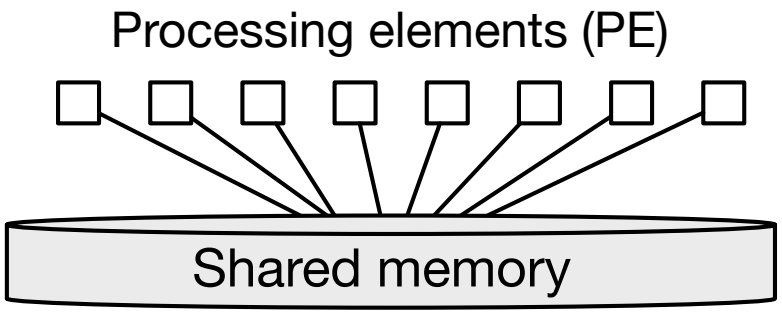


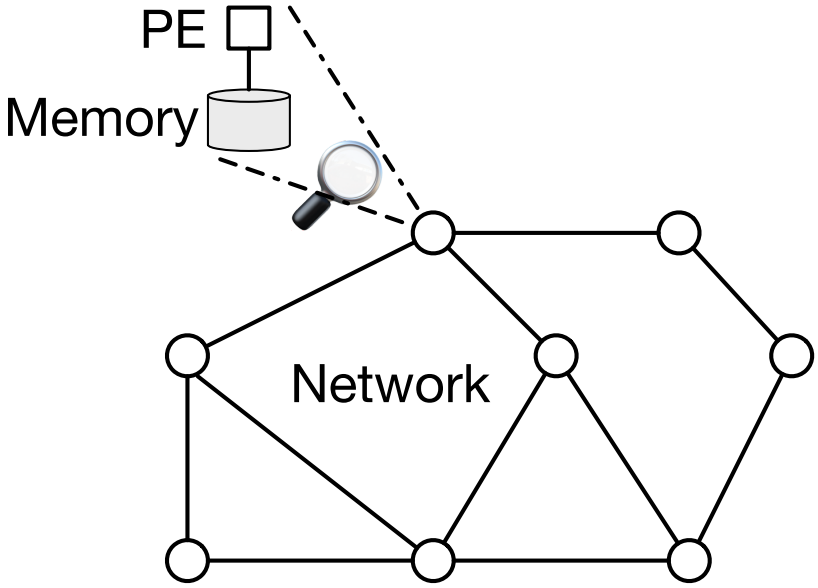
**Alonzo Church**

Typical model for distributed computing



# Examples of Communication Media



# Limitations Faced by Distributed Computing: Undecidability  + Uncertainty 

**Sources of uncertainties:**

- Spatial: communication network
- Temporal: clock drifts (asynchrony, load, etc.)
- Failures (transient, crash, malicious, etc.)
- Selfish behavior (game theory)
- …

**Several Turing machines are weaker than one!**

# 2.18.x Courses

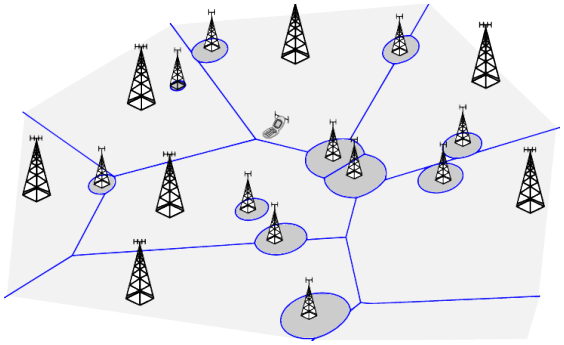
- 2.18.1 Distributed algorithms on networks
  - ✓ Spatial issues: locality

- 2.18.2 Distributed algorithms on shared memory
  - ✓ Temporal issues: asynchrony and failures

# Symmetry Breaking

- Leader election
- Consensus
- Coloring
- Graph problems
- Etc.

**Applications :**



**Frequency assignments**

**Distributed data-bases consistency**

# 2.18.1 Course (2024-2025)

1. Pierre Fraigniaud *(CNRS and Université Paris Cité)*

2. Mikaël Rabie *(Université Paris Cité)*

firstname.lastname@irif.fr

# Prerequisites

Basic knowledge in:

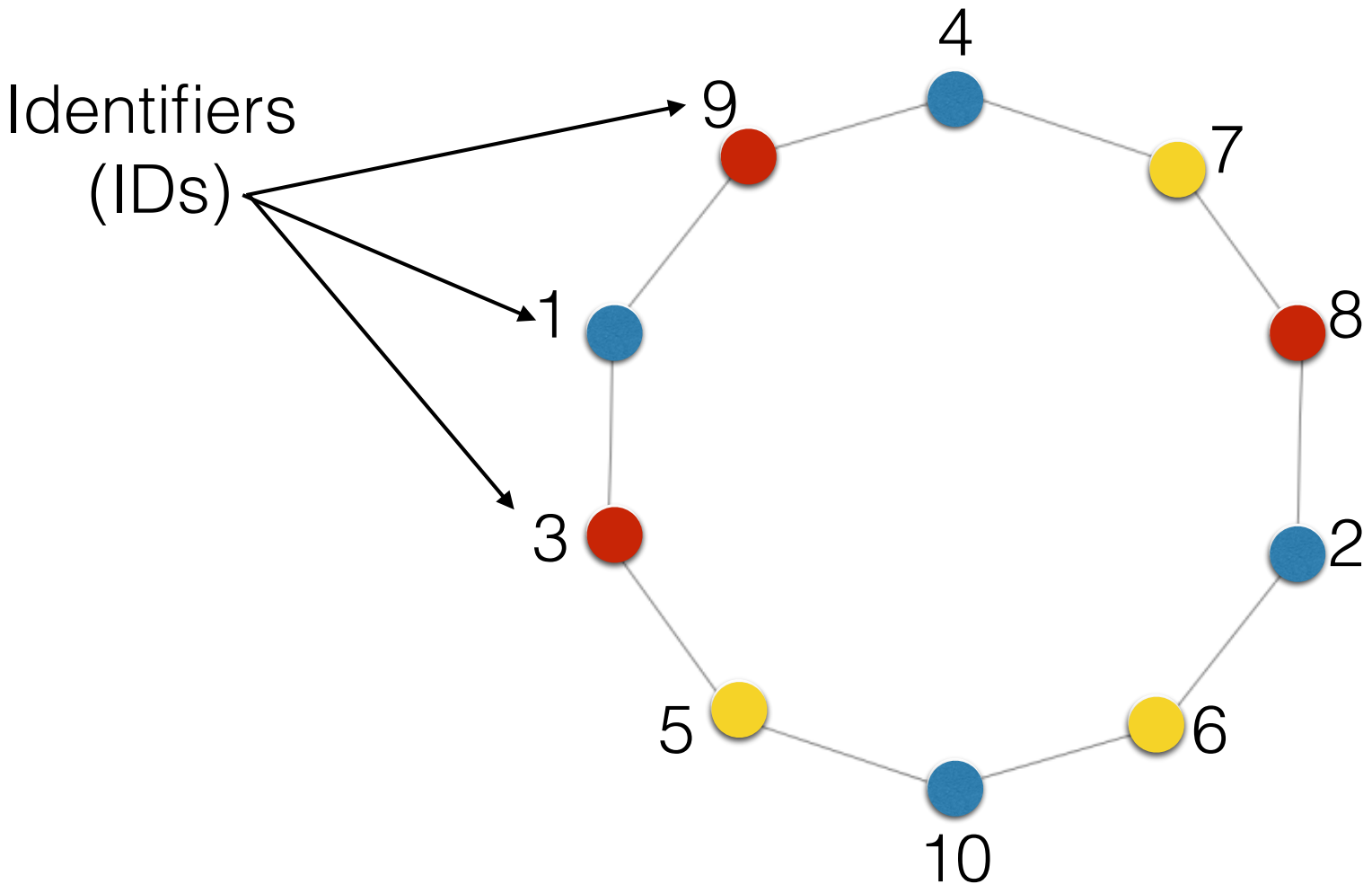
- Graph theory
- (Sequential) algorithm design and analysis
- Elements of probability theory

# Roadmap Lecture 1

- 3-coloring the $n$-node cycle $C_n$
- $(\Delta + 1)$-coloring $n$-node graphs of maximum degree $\Delta$

# 3-coloring $C_n$

# 3-coloring the $n$-node cycle $C_n$




We assume synchronous rounds,
and no failures

# Color Reduction

## (assuming IDs in $\{1,\ldots,n\}$)

- MyColor ← MyID
- For $i = n$ down to $4$ do
  - Send MyColor to neighbors
  - Receive colors from neighbors
  - if Mycolor $= i$ then pick smallest color in $\{1,2,3\}$ distinct from colors of neighbors

**Complexity:** $O(n)$ rounds

# Color Reduction
## (assuming arbitrary IDs)

- MyColor ← MyID
- Repeat
  - Send MyColor to neighbors
  - Receive colors from neighbors
  - if Mycolor is locally maximum then
    - pick smallest color in $\{1,2,3\}$ distinct from colors of neighbors
    - send MyColor to neighbors and terminate

**Complexity:** $O(n)$ rounds

# Logarithms

- Unless specified otherwise, all logarithms in these lectures are in base 2

$$\text{For every } x > 0,\ \log_2 x = \frac{\ln x}{\ln 2}$$

- $k$-bit binary strings encode integers between 0 and $2^k - 1$ (or between $a$ and $a + 2^k - 1$ for any $a \in \mathbb{Z}$)

- For every $n > 0$, encoding all integers in $[0,n-1]$ uses $\lceil \log_2 n \rceil$ bits.

- Encoding $n > 0$ values uses $\lceil \log_2 n \rceil$ bits.

# A Super Fast Algorithms

**Theorem** (Cole and Vishkin, 1986) There exists a distributed algorithm for $3$-coloring $C_n$ performing in $O(\log^\star n)$ rounds.

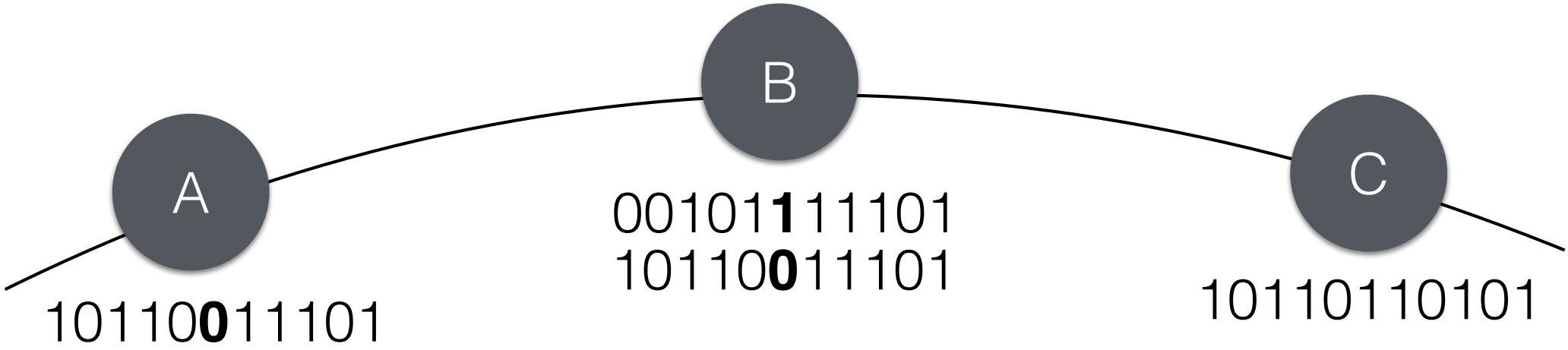
Iterated logarithms:

- $\log^{(0)} x = x$ $\quad\quad$ $\log^{(k+1)} x = \log \log^{(k)} x$
- $\log^* x$ = smallest $k$ such that $\log^{(k)} x < 1$
- $\log^* 10^{100} = 5$

# Cole-Vishkin Algorithm

Initial color = ID
Express colors in binary

Assume: n is known, and
consistent sens of direction




new = (position,bit) = (5,1) = 1011

(p',b') (p,b)

p ≠ p' ⇒ proper coloring

p = p' ⇒ b ≠ b' ⇒ proper coloring

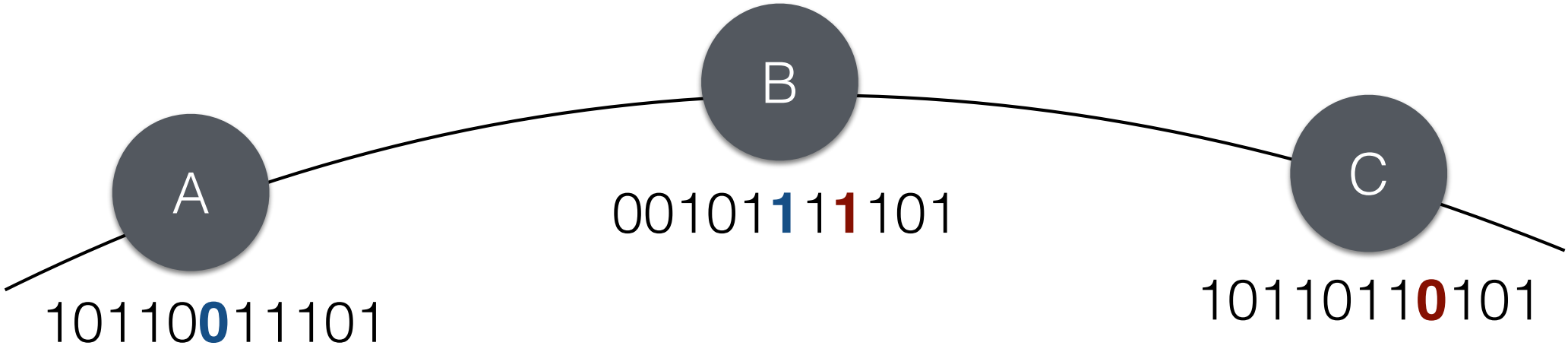
# Number of iterations

- $k$-bit colors ⇒ new colors on $\lceil \log_2 k \rceil + 1$ bits
- $\log^\star n + O(1)$ rounds to reach colors on 3 bits
- 8 colors down to 3 colors in 5 rounds
- Total number of rounds = $\log^\star n + O(1)$

In fact: 6 colors down to 3 colors in 3 rounds

❑

# Speeding up CV's Algorithm




- Every node can simulate 2 rounds in just 1 round
- left round + right round ➡ implemented in 1 round
- Total number of rounds = $\frac{1}{2}\log^{\star} n + O(1)$ ❑

# Question

Is there a distributed algorithm for 3-coloring the n-node ring performing in less that $\Omega(\log^\star n)$ rounds?

**Theorem** (Linial, 1992) Any distributed algorithm for $3$-coloring $C_n$ has round-complexity at least

$$\frac{1}{2}\log^\star n - O(1)$$

Dijkstra Prize 2013

*This will be established later in the course*

$(\Delta + 1)$-coloring arbitrary $n$-node graphs with maximum degree

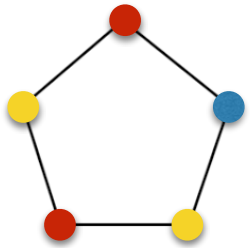
$$\Delta = \max_{v \in V(G)} \deg(v)$$

# $(\Delta+1)$-coloring

$\Delta$ = maximum node degree of the graph

$(\Delta+1)$-coloring = assign colors to nodes such that every pair of adjacent nodes are assigned different colors.

**Lemma** Every graph is $(\Delta+1)$-colorable



**Theorem** (Brooks, 1941)
Every graph $G$ is $\Delta$-colorable, unless $G$ is a complete graph, or an odd cycle.

**Lemma** $(\Delta+1)$-coloring can be sequentially computed by a simple greedy algorithm treating each node individually.

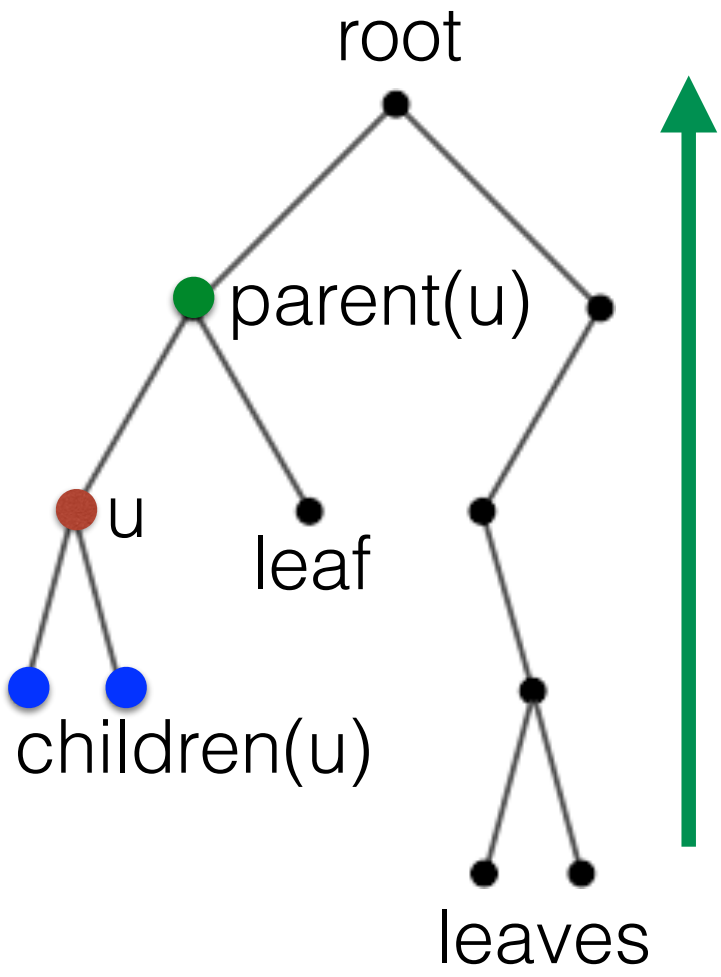
# Remark

Let $k \geq \Delta + 1$

If there exists a $t$-round $k$-coloring algorithm then there exists a $(\Delta + 1)$-coloring algorithm running in $t + \left(k - (\Delta + 1)\right)$ rounds.

# 3-Coloring Rooted Trees

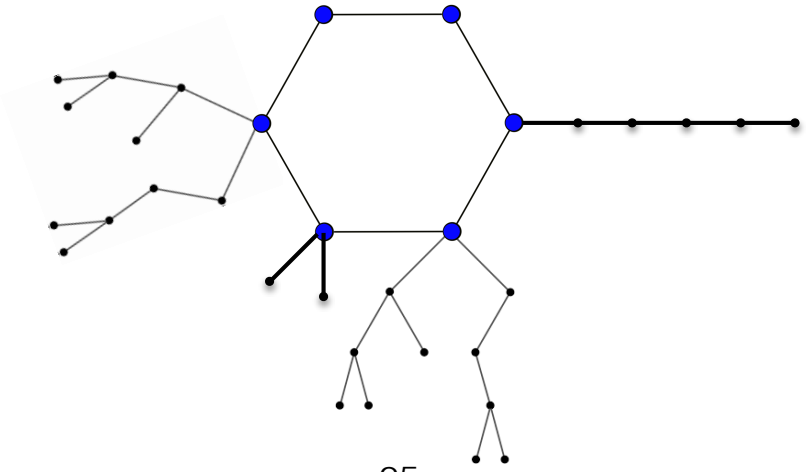
- Apply C&V with parent for $O(\log^\star n)$ rounds, to 6-color the tree
- For $i = 6$ down to $4$ do
  - adopt color of parent
  - recolor nodes colored $i$ with a color in $\{1,2,3\}$



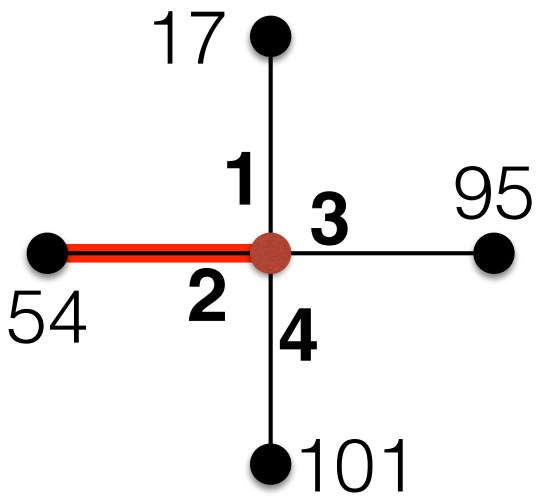
# 1-Factors

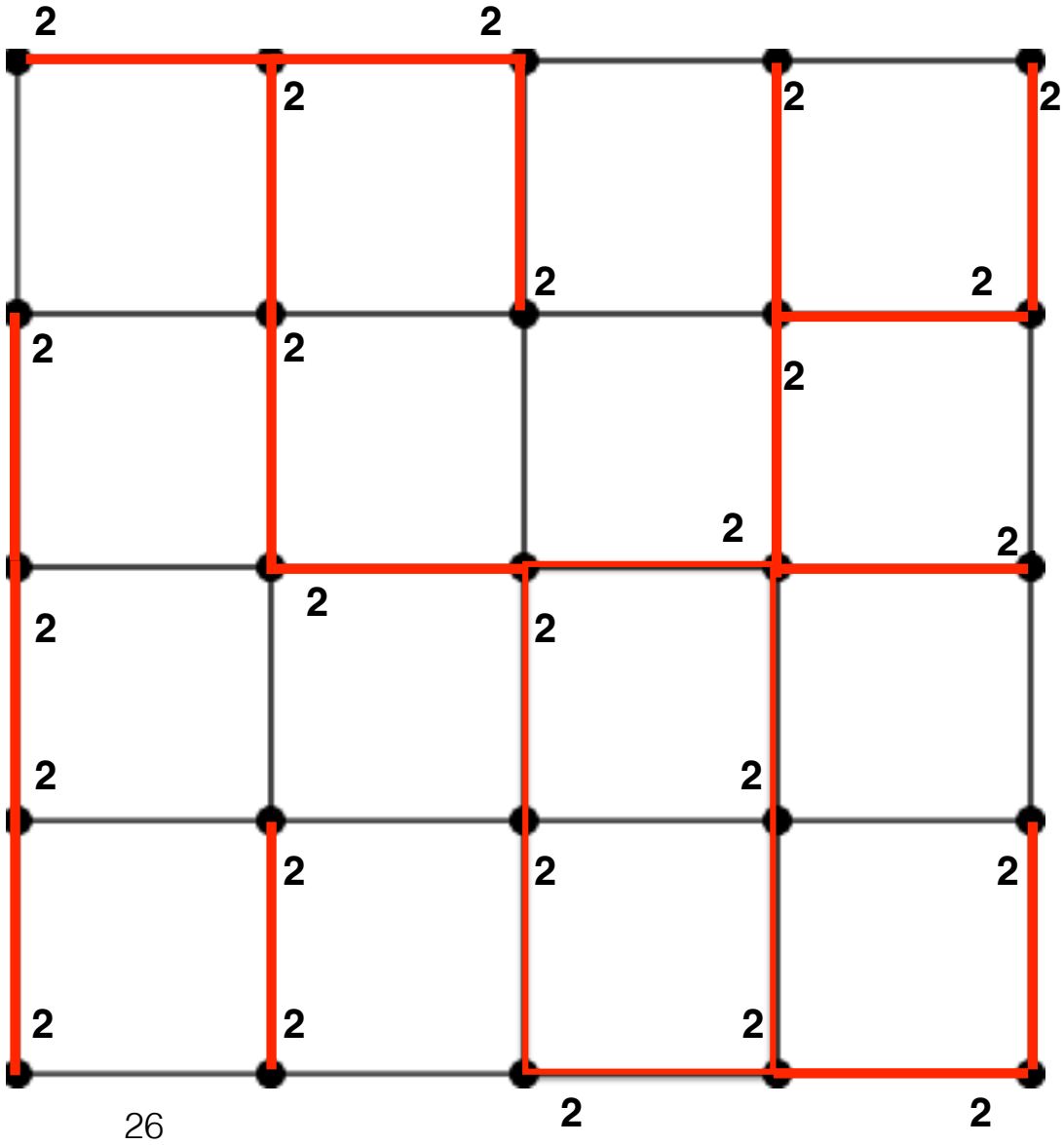
- Let $G = (V, E)$ be a graph
- Assume each node $v \in V$ selects one of its incident edges
- Let $F \subseteq E$ be the set of selected edges

**Claim** $F$ is a collection of « pseudo-trees » of the form

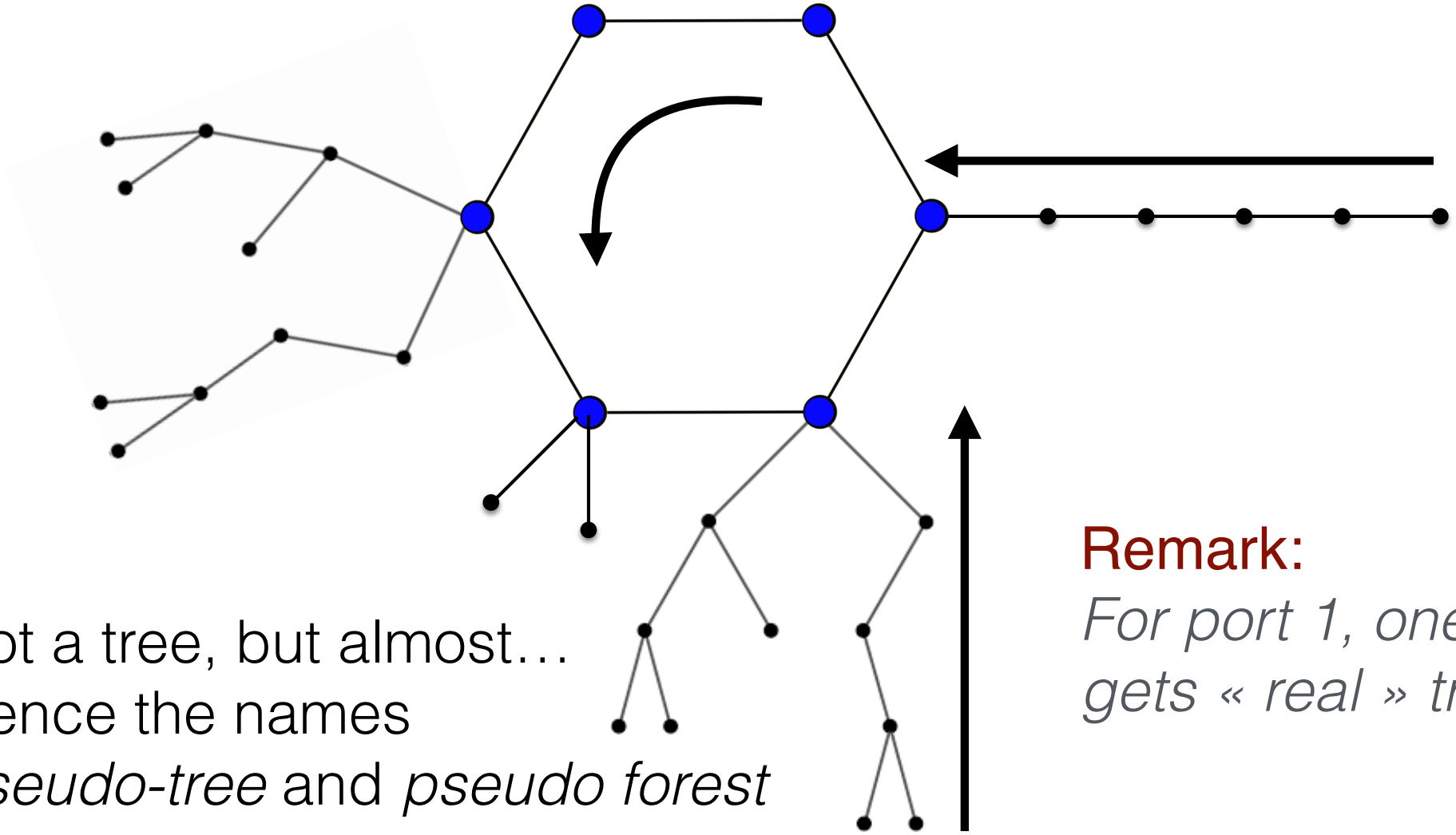


# Pseudo-Forest Decomposition



# A Connected Component



Not a tree, but almost…
Hence the names
*pseudo-tree* and *pseudo forest*

**Remark:**
*For port 1, one gets « real » trees*

# Coloring with $3^{\Delta}$ colors in $O(\log^{\star} n)$ rounds

- Every node $u$ orders its incident links from 1 to $\deg(u)$ according to the IDs of its neighbors
- This results in $\Delta$ pseudo-forests $F_1, \ldots, F_{\Delta}$
- Color each pseudo tree in each pseudo forest in parallel, in $O(\log^{\star} n)$ rounds
- Each node gets a color $c(u) = (c_1(u), \ldots, c_{\Delta}(u))$ where $c_i(u) \in \{1,2,3\}$, hence $3^{\Delta}$ colors.

End Lecture 1