Distributed Algorithms (3)

- Last lecture is next week 10/12
  we will have a test 50% (50% Exam) in January

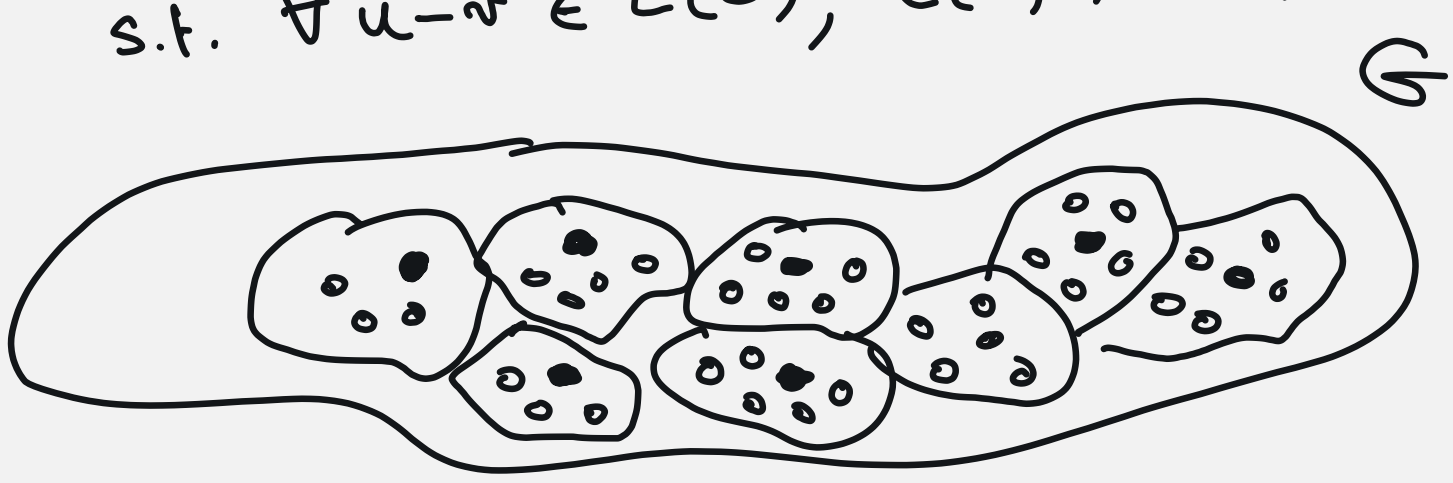
TODAY

Chapter 7 COLORING

We will on the LOCAL model.
We care about TIME.
We want to partion the vertices into "colors" that are independent sets.

A $\boxed{k\text{-coloring}}$ for a graph G

$c : V(G) \longrightarrow \{0, 1, \dots, k-1\}$
s.t. $\forall u - v \in E(G)$, $c(u) \neq c(v)$.

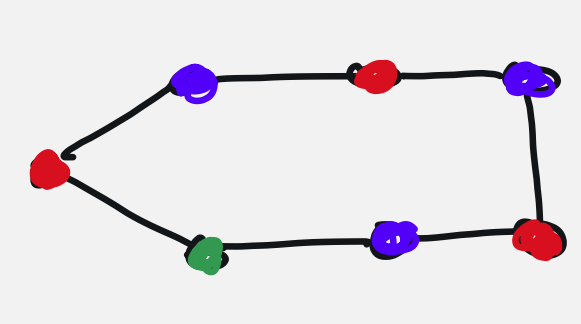


This is a "breaking symetry" problem

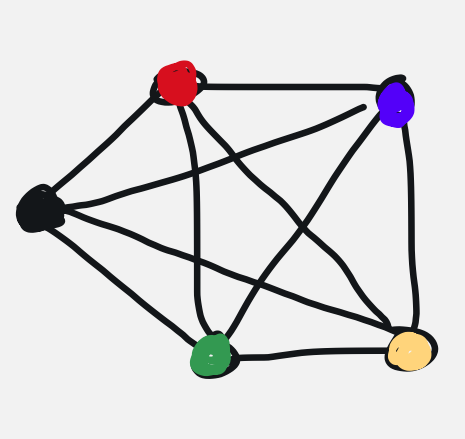
Computing the minimum k s.t. G has a k-coloring is NP-hard.

Goal: Compute "quickly" a coloring with "few" colors.

- "quickly" means $\ll$ diam(G).
  typically $O(1)$, $O(\log n)$ rounds.
- "few" colors means $\Delta(G)+1$
  ↑ max. deg of G.



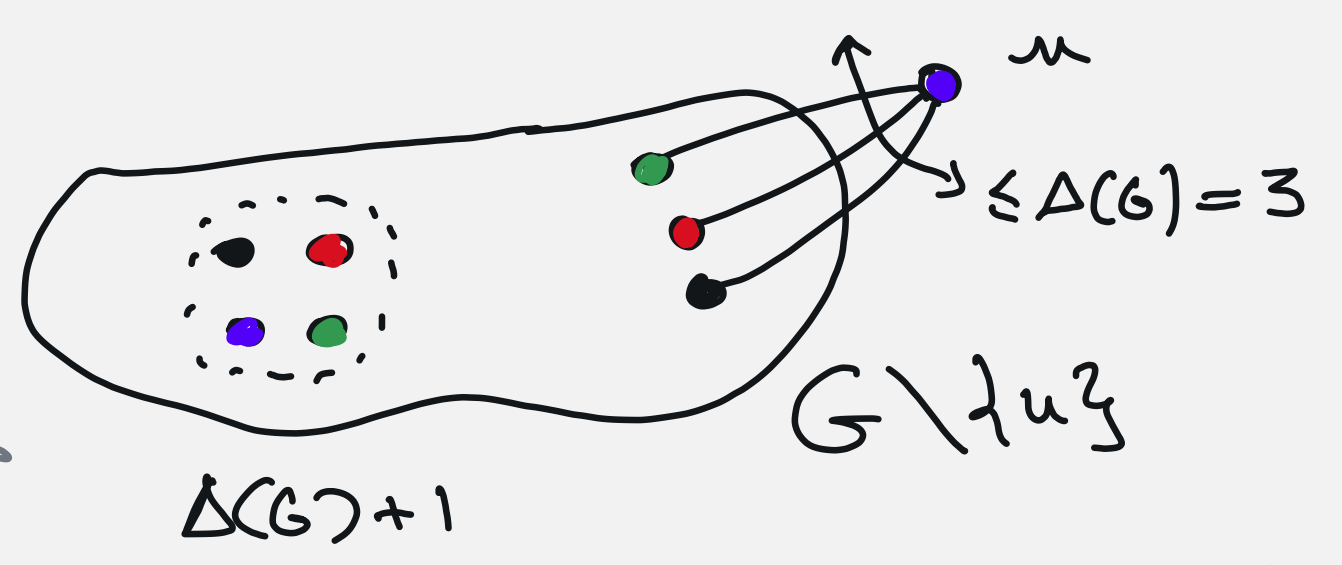
$\Delta(G) = 2 \Rightarrow 3$-coloring



$\Delta(K_m) = m-1 \Rightarrow \Delta(G)+1$
we need m colors

For planar graphs we can always find a 4-coloring.
It is always possible to color a graph with $\Delta(G)+1$ colors.

by induction on the number of vertices



In sequential, we use a greedy algo.

GREEDY_COL(G)

```
1. ∀u ∈ V(G), c(u) := ID(u) ∈ {0, ..., n-1}
2. For each u, successively,
       c(u) := FirstFree(c(N(u)))
```

↖ this is a valid coloring

← a new coloring

↑ set of neigh.

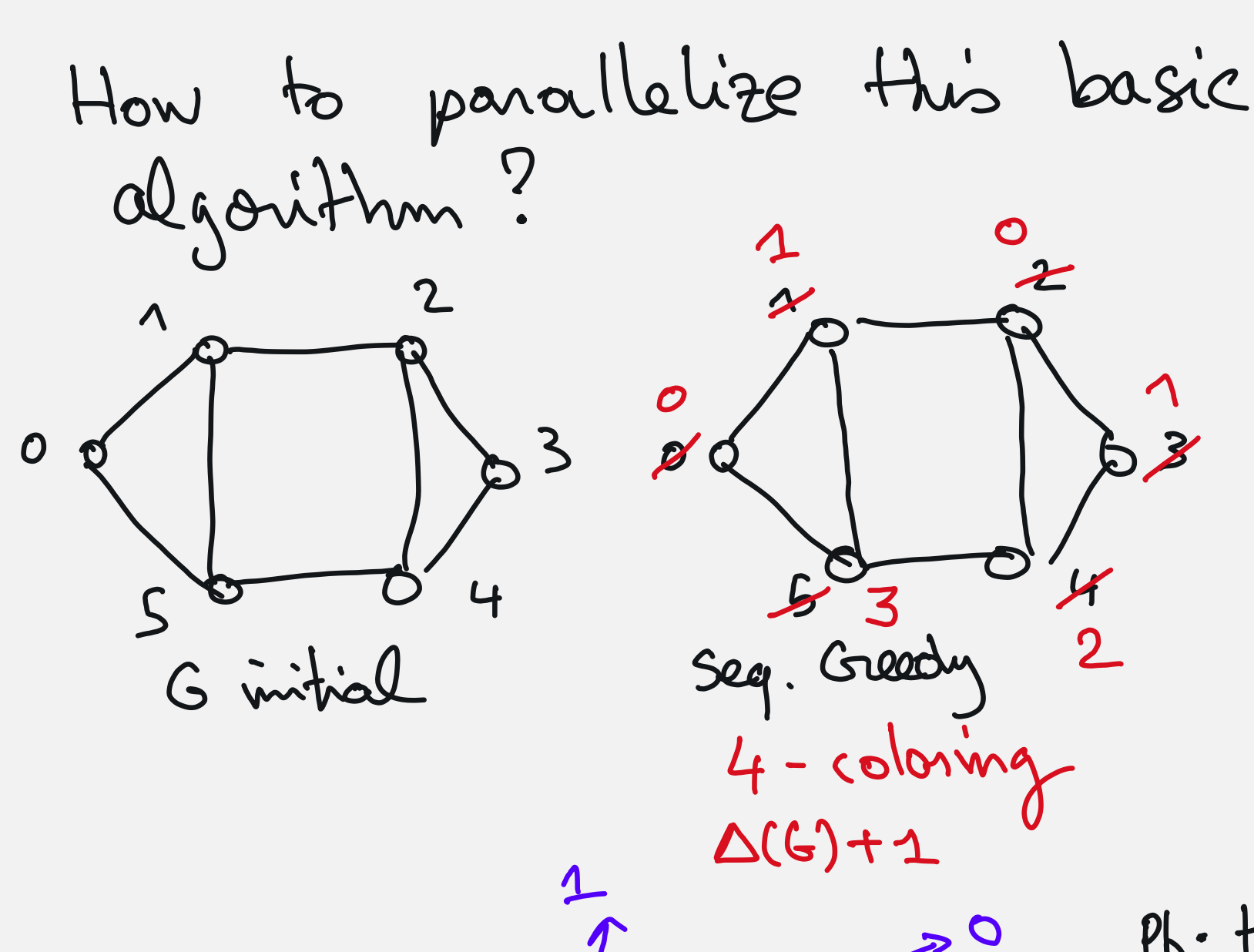
↑ set of colors of all neighbors of u.

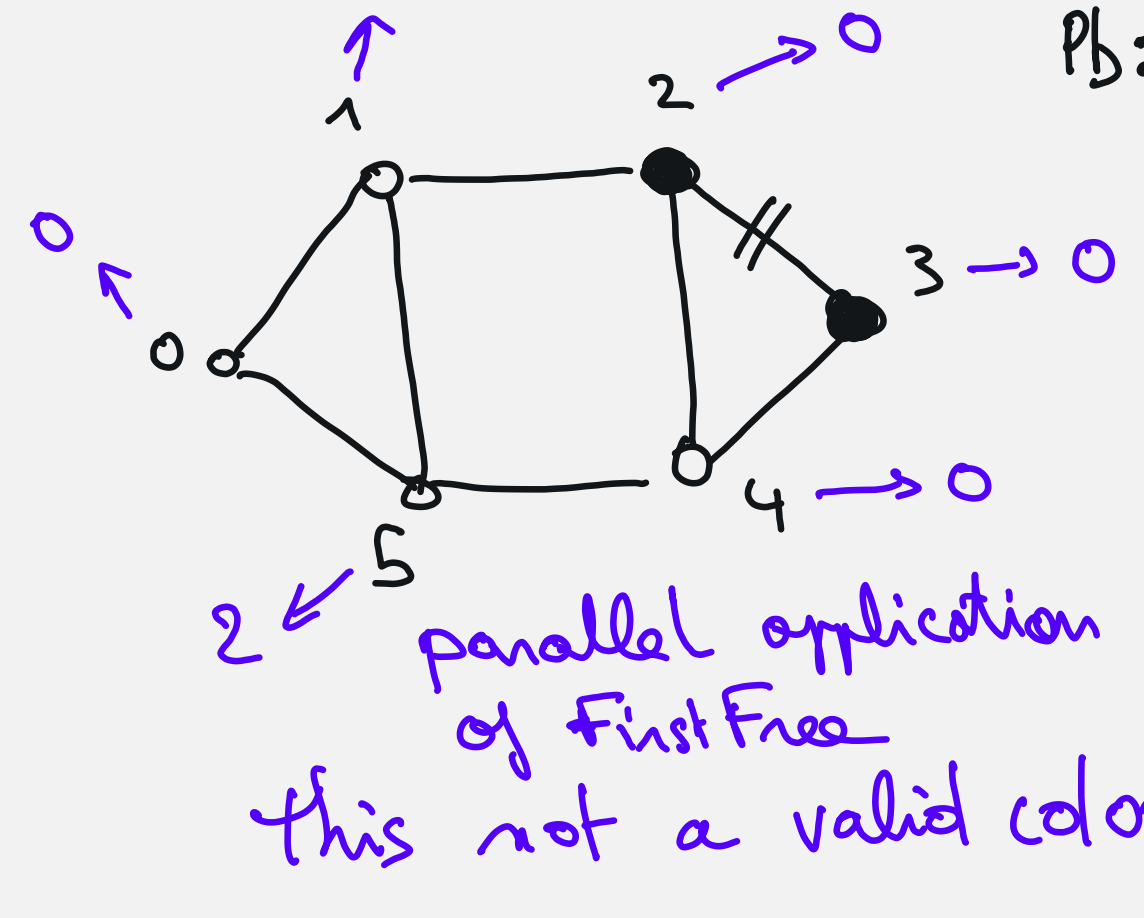
$c(u) \in \{0 \dots\dots \deg(u)\} \leq \Delta(G)$

Function $\mathrm{FirstFree}(X) := \min(\mathbb{N} \setminus X)$
$X \subseteq \mathbb{N}$ (the smallest int. ≥ 0 that is $\notin X$)

Ex: $\mathrm{FirstFree}(\{0,1,2,4,8\}) = 3$
$\mathrm{FirstFree}(X) \in \underbrace{\{0,1,2,\dots,|X|\}}_{|X|+1} \leq |X|$
$\mathrm{FirstFree}(c(N(u))) \leq \deg(u) \leq \Delta(G)$.

How to parallelize this basic greedy algorithm?





Pb: two neighbors can take the same decision.

parallel application of FirstFree
this not a valid coloring!

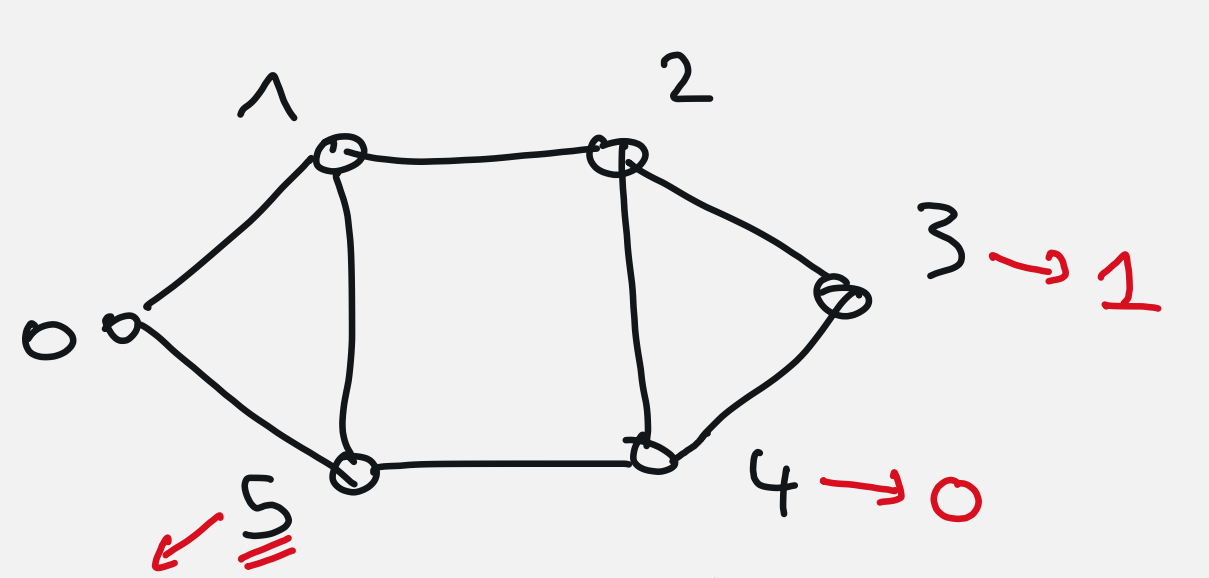
Solutions with a distributed algorithms

Idea: try to reduce as quick as possible the number of colors (starting with an n-coloring)

Reduce Color (c)

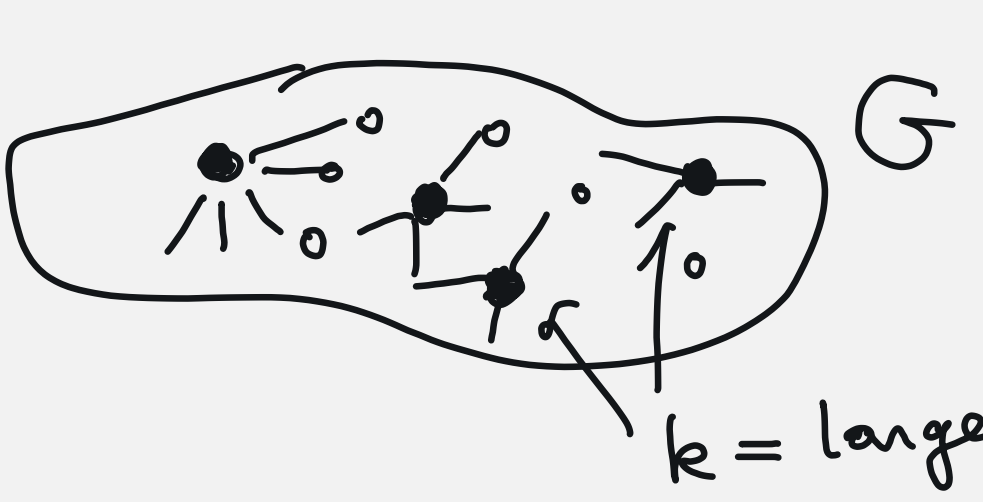
(code of u) ↑ some coloring

```
• New Round
• SEND ALL (c(u))
• X := ⋃_{v∈N(u)} {RECEIVE(v)}        (X = c(N(u)))
[• c(u) := FirstFree(X)]              Bad idea   0 ← u ! v → 0
• If c(u) > max(X), c(u) := FirstFree(X).
```

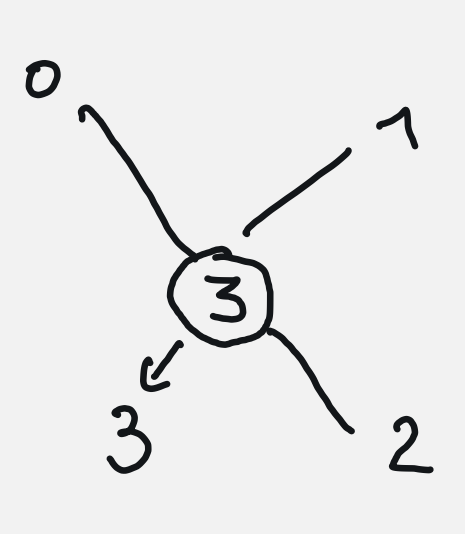


Reduce Color
Reduce Color
Reduce Color
⇒ 3-coloring

We can apply ReduceColor(c) to remove the largest current color in G.

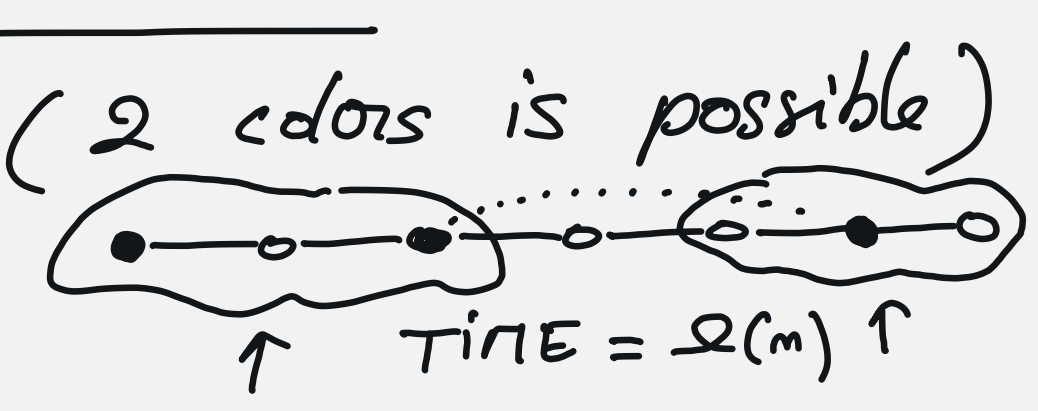


k = largest number
(k+1-coloring)
↳ FirstFree $\leq \Delta(G)$
$\Delta(G)+1$-coloring
if we apply ReduceColor at most $n - \Delta(G)$.



COLORING FOR TREES (AND CYCLES)

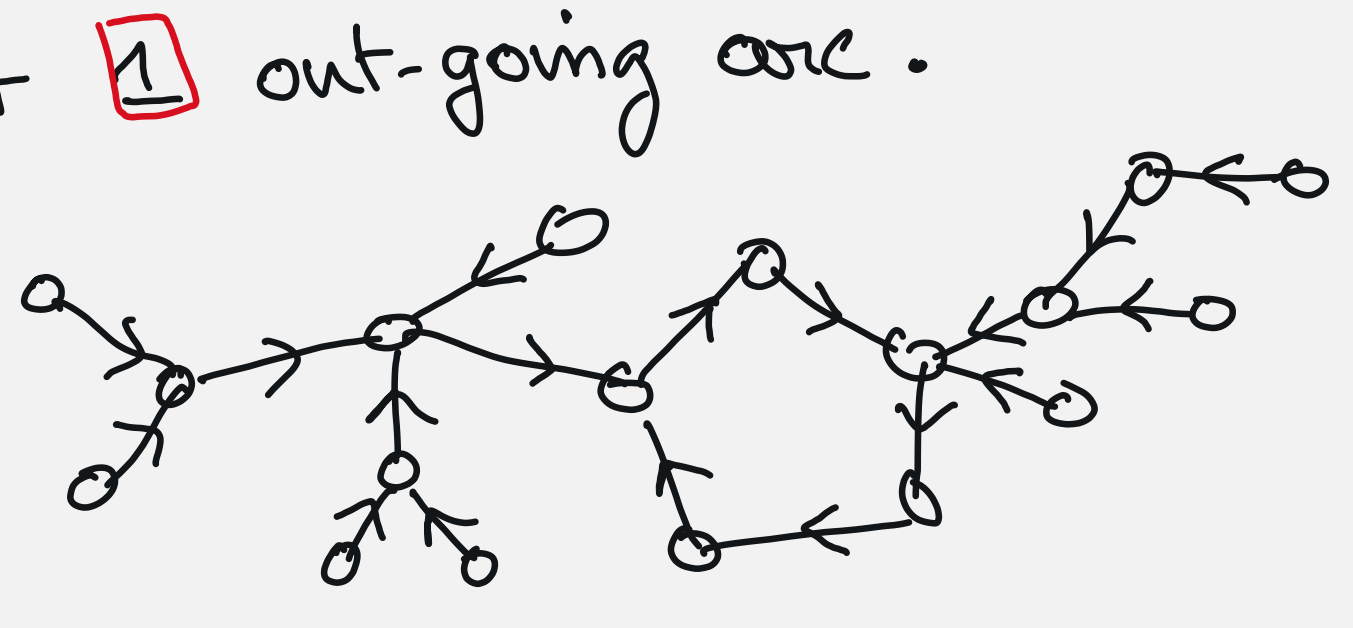
$\Delta(G)+1$ colors (2 colors is possible)

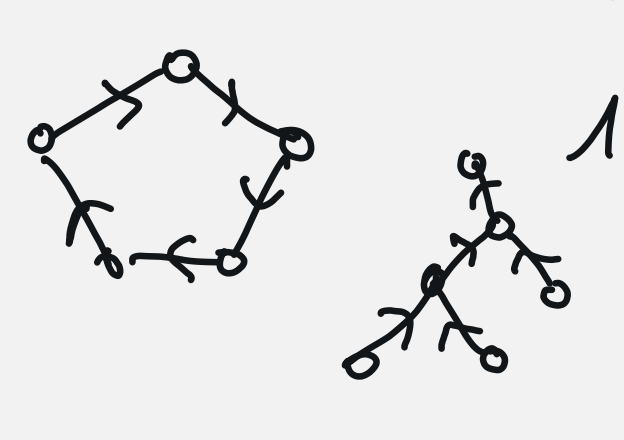


↑ TIME = $\Omega(n)$ ↑

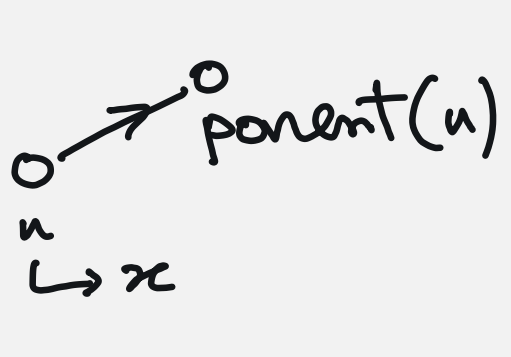
We will see a very efficient algorithm for trees and cycles, more generally 1-orientation

A graph is a "1-orientation" if its edges can be oriented s.t. every vertex has at most 1 out-going arc.





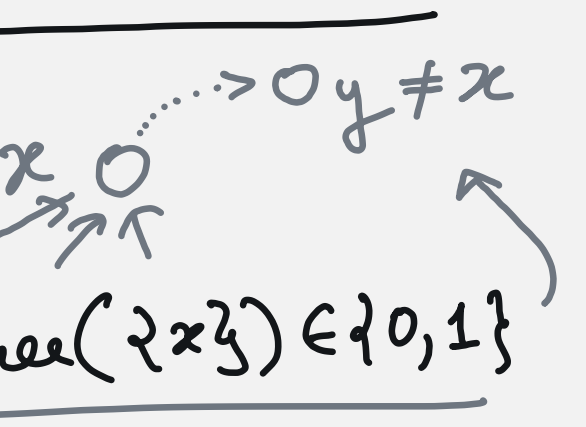
Assume $u \longmapsto \mathrm{Parent}(u)$
$n = V(G)$

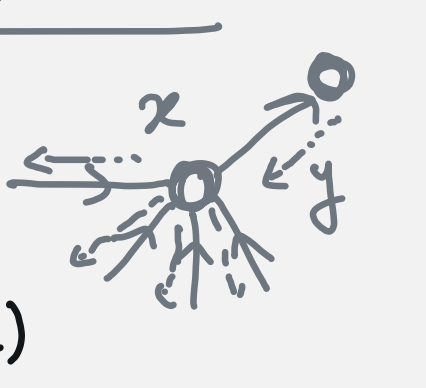


COLOR6

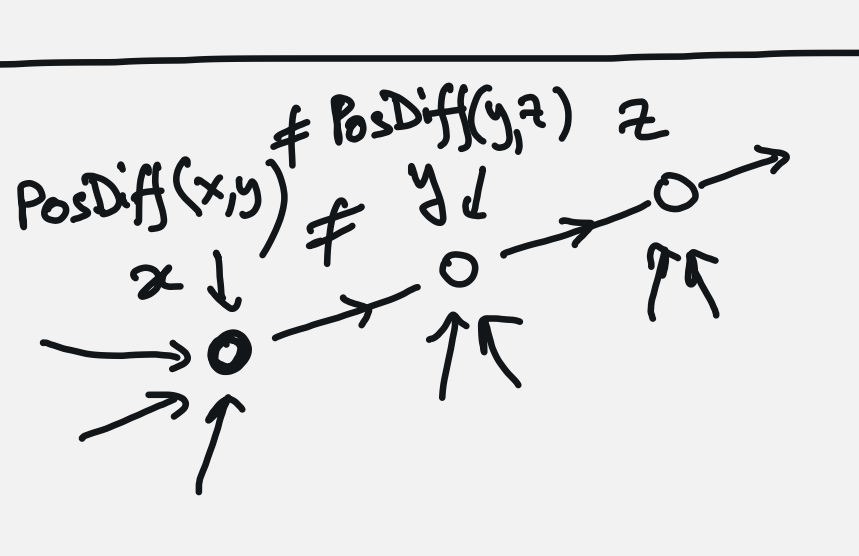
(code of u)

```
        {0,...,n-1}
1. x := ID(u), ℓ := ⌈log₂ n⌉
2. If parent(u) = ⊥, y := FirstFree({x}) ∈ {0,1}
3. Repeat:
     a. New Round
     b. SEND(x, v)  ∀v ∈ N(u) \ parent(u)
     c. If parent(u) ≠ ⊥, y := RECEIVE(parent(u))
     d. x := PosDiff(x, y)
     e. ℓ' := ℓ, ℓ := 1 + ⌈log₂ ℓ⌉
   Until ℓ = ℓ'
4. COLOR(u) := x
```





in general $1 + \lceil \log \ell \rceil \ll \ell$.
$\ell = 1024 = 2^{10}$
$1 + \lceil \log 1024 \rceil = 1 + 10 = 11 \ll 1024$.
⚠ $1 + \lceil \log 3 \rceil = 3$



Function PosDiff(x, y) $x, y \in \mathbb{N}$

$\mathrm{PosDiff}(x,y) = 2p + \mathrm{bin}(x)[p]$
where p is any position where bin(x) and bin(y) differ.
↳ the smallest

binary writing of x
$x = \overset{4\,3\,2\,1\,0}{10110}$
bin(x)
$\mathrm{bin}(x)[2] = 1$

Ex: $\mathrm{PosDiff}(10, 4) = 2 \times 1 + 1 = 3$

(10) $\overset{3\,2\,1\,0}{1010_2}$
(4) 100_2
p = 1

$\mathrm{PosDiff}(6, 22) = \mathrm{PosDiff}(110_2, 10110_2) = 2 \cdot 4 + 0 = 8$

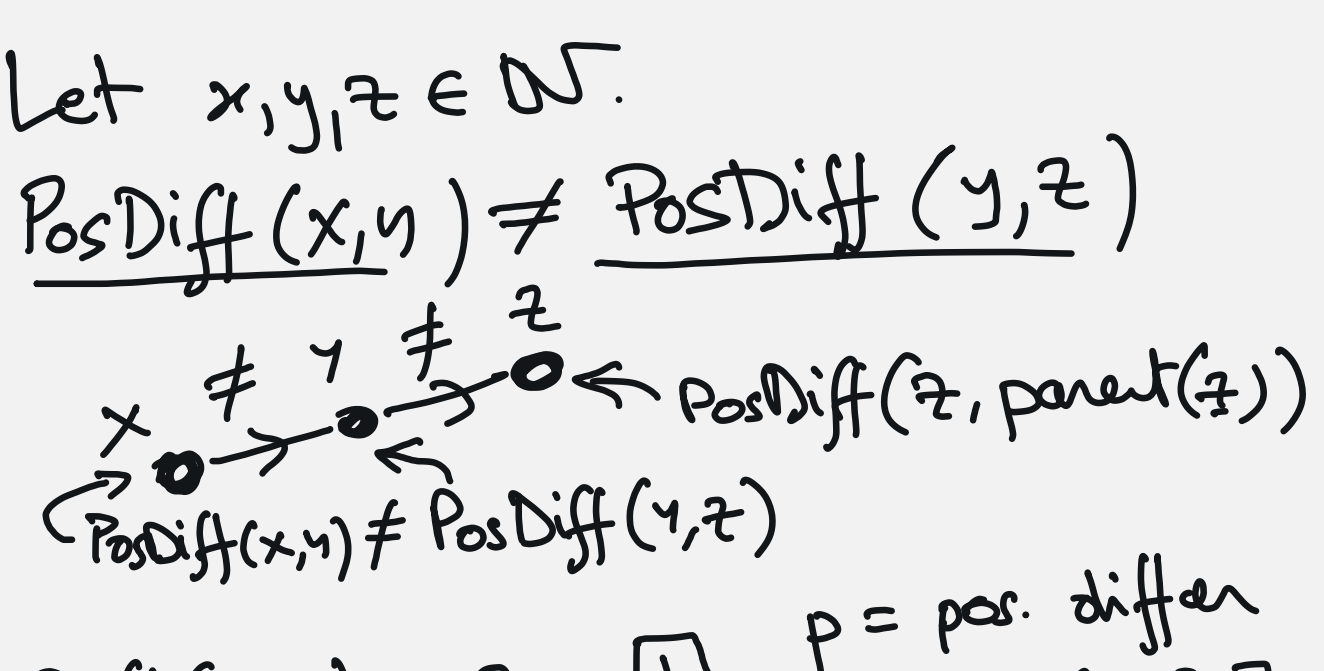
x $\overset{4\,3\,2\,1\,0}{00110}$
10110

$\mathrm{PosDiff}(4, 0) = \mathrm{PosDiff}(\overset{2\,1\,0}{100_2}, 000) = 2 \cdot 2 + 1 = 5$

$\mathrm{PosDiff}(4 \cdot i, 2) = \mathrm{PosDiff}(\mathrm{bin}(i) \cdot \overset{1\,0}{00_2}, 10_2)$
$= 2 \cdot 1 + 0 = 2$

$\mathrm{PosDiff}(0, \underline{2i+1}) = \mathrm{PosDiff}(0_2, \mathrm{bin}(i) \cdot 1_2)$
$= 2 \cdot 0 + 0 = 0$

Property. Let $x, y, z \in \mathbb{N}$.
If $x \neq y$, $y \neq z$, $\mathrm{PosDiff}(x,y) \neq \mathrm{PosDiff}(y,z)$



Proof. $\mathrm{PosDiff}(x,y) = 2p + \boxed{b}$
$\mathrm{PosDiff}(y,z) = 2p' + \boxed{b'}$ $\{0,1\}$

p = pos. differ
$b = \mathrm{bin}(x)[p]$
$b' = \mathrm{bin}(y)[p']$

Suppose that $\mathrm{PosDiff}(x,y) = \mathrm{PosDiff}(y,z) = v$
$\Rightarrow p = p'$ and $b = b'$.

$b = v \bmod 2$
$p = \lfloor v/2 \rfloor$

If $p = p'$, $b = \mathrm{bin}(x)[p]$,
$b' = \mathrm{bin}(y)[p'] = \mathrm{bin}(y)[p] = b' \neq b$

We said that x and y DIFFER at position p.
⇒ this a contradiction. $\mathrm{PosDiff}(x,y) \neq \mathrm{PosDiff}(y,z)$ ∎

TIME ?
# COLORS ?

$x_0(u) = ID(u)$
$\ell_0 = \lceil \log n \rceil$
$\ell_1 = 1 + \lceil \log \ell \rceil = 1 + \lceil \log \lceil \log n \rceil \rceil$,
$= O(\log \log n)$
$\ell_2 = 1 + \lceil \log \ell \rceil = 1 + \lceil \log \downarrow \rceil = O(\log \log \log n)$
⋮
$\ell_i = O(\log^{(i+1)} n)$

Property: $\forall i > 0$, if $\ell_{i-1} > 3$, then $\ell_i < \ell_{i-1}$
else $\ell_i = \ell_{i-1}$.
$1 + \lceil \log 2 \rceil = 1 + \lceil 1 \rceil = 2$ ↗

Property: $\underbrace{|\mathrm{bin}(x_i(u))|}_{\#\text{ bits}} \leq \ell_i$.

Property: At the end of COLOR6, $x_i(u) \in \{0 \dots 5\}$
⇒ 6-COLORING.

When COLOR6 stops $\ell_{i-1} = \ell_i \leq 3$.
$x_i(u) = \mathrm{PosDiff}(x_{i-1}(u), x_{i-1}(v)) = 2p + b$
$p \in \{0, 1, 2\}$ $\ell_{i-1} \leq 3$ $\overset{2\,1\,0}{000}$
$b \in \{0, 1\}$
$2p + b \leq 2 \cdot 2 + 1 = 5$ ∎

Property: If $\ell_{i-1} > 3$, $\ell_i \leq 2 + \lceil \log^{(i+1)} n \rceil \leq 3$

DEF. $\log^* n = \min\{i : \log^{(i)} n \leq 1\}$ Find i in $\log^* n$

⇒ this is the smallest $i \in \mathbb{N}$
s.t. $n \leq \left.2^{2^{\cdot^{\cdot^{2}}}}\right\}i$

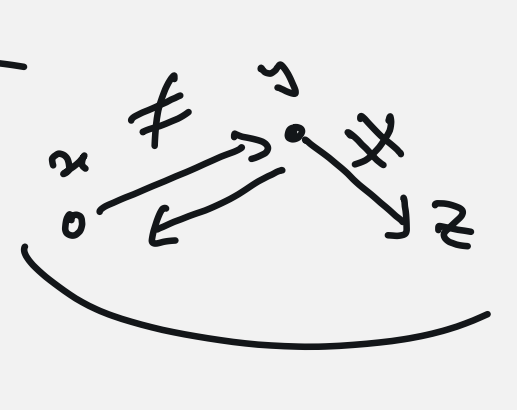
$\log^*(2^{65536}) = \log^*(2^{2^{16}}) = \log^*(2^{2^{2^{2^2}}}) = 5$

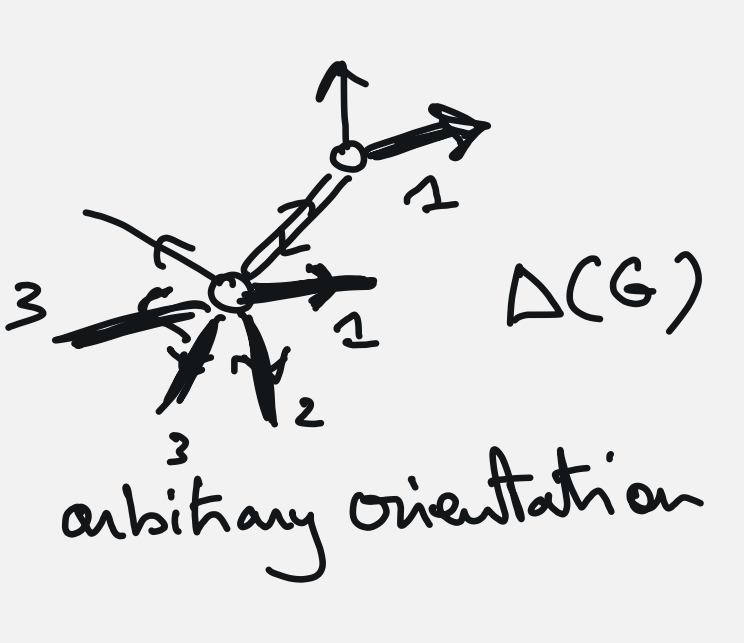
NB: # elem. particules in Univers is $\leq 2^{400}$

Lemma 7.1 Algorithm COLOR6 computes a 6-coloring in time $\leq \log^* n$ (≤5) in every graph having 1-oriented with n vertices.

There is a trick that we can apply to the end of COLOR6 so that it produces 3 colors. (ReduceColor 3 times to remove color 5, 4, 3)
$\log^* n + 3$ rounds ⇒ 3-COLORING.

Application to every graph G





$\overset{x}{c}(u) \to (x_1, x_2, \dots x_{\Delta(G)})$
↓ $\log^* n$ rounds
$(c_1, c_2, c_3, \dots c_\Delta)$ $c_i \neq c'_i$
i ↗
$(c'_1, c'_2, c'_i, c'_\Delta)$

$c_i \in \{0, 1, 2\}$
⇒ $\boxed{3^\Delta - \text{COLORS}}$

Applying the $\log^* n + O(1)$ to a $\Delta(G)$-orientation we get a 3^Δ coloring.
Applying ReduceColor $(3^\Delta - \Delta)$ Times we get $(\Delta+1)$-COLORS.
in total Time $\log^* n + O(3^\Delta)$ rounds $\forall G$

END.