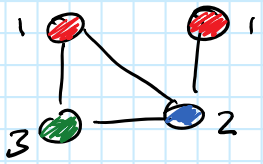
Vertex Coloring

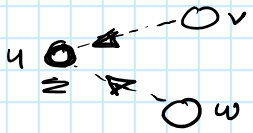
Wednesday, May 07, 2014 10:56 PM



usually goal:
use as few colors as possible

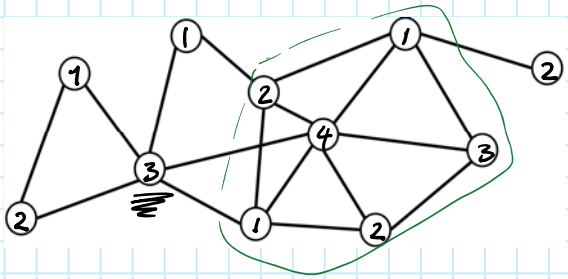
chromatic number $\chi(G)$
colors of an optimal coloring
$\Rightarrow$ determining / approx $\chi(G)$ is NP-hard

possible application
wireless networks



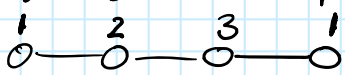
Sequential Alg.

greedy:



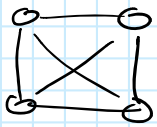
$\Delta = 6$

example where greedy is not optimal:



How good is greedy?

Δ : largest degree of any node

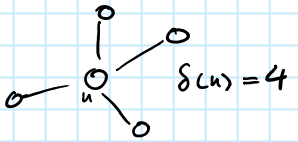


$\Delta = 3$

$\chi = 4$

Claim : Greedy uses $\leq \Delta + 1$ colors.

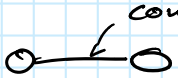
Proof :



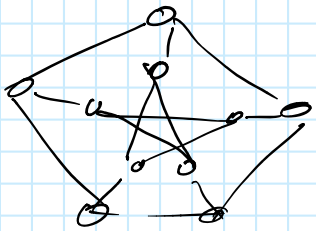
color of $u \leq 5 = \delta(u) + 1$

Can we achieve $\Delta+1$ colors with a distributed algorithm?

Distr. Alg:
nodes with IDs



Ideas
one starts, neighbors choose differently, repeat
↑ smallest degree



or the one with the smallest ID

Procedure 1.2 First Free

Require: Node Coloring {e.g., node IDs as defined in Assumption 1.2}
Give v the smallest admissible color {i.e., the smallest node color not used by any neighbor}

IDs = coloring

Choose new color if you're the smallest ID among neighbors

↑ to no two neighbors choose a new color

undecided

Synchronous Distributed Alg.

Nodes operate in rounds

In each round, each node/processor does (in parallel)

1. do some local computation (reasonable compl.)
2. send messages to all neighbors (of reasonable size)
3. receive messages from all neighbors

Remark:

reordering 1.-3. is fine

Algorithm 1.3 Reduce

1: Assume that initially all nodes have ID's (Assumption 1.2)
2: **Each node** v executes the following code
3: node v sends its ID to all neighbors
4: node v receives IDs of neighbors
5: **while** node v has an uncolored neighbor with higher ID **do**
6: node v sends "undecided" to all neighbors ← not necessary
7: node v receives new decisions from neighbors
8: **end while** (incl. color)
9: node v chooses a free color using subroutine **First Free** (Procedure 1.2)
10: node v informs all its neighbors about its choice

Time Complexity

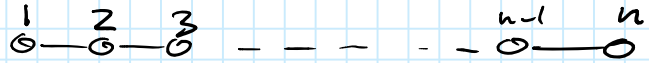
number of rounds

Time Compl. of Alg. 1.3 ?

at least one node per round gets a color

⇒ time compl. $O(n)$

It can be that bad:

- complete graph
- 

Message Complexity

total # of msg.

Alg. 1.3 : $O(m \cdot n)$

If we remove line 6 : $O(m)$

Remarks

- quite trivial → quite slow
- improve it? use randomization

⇒ instead general graphs → look at trees