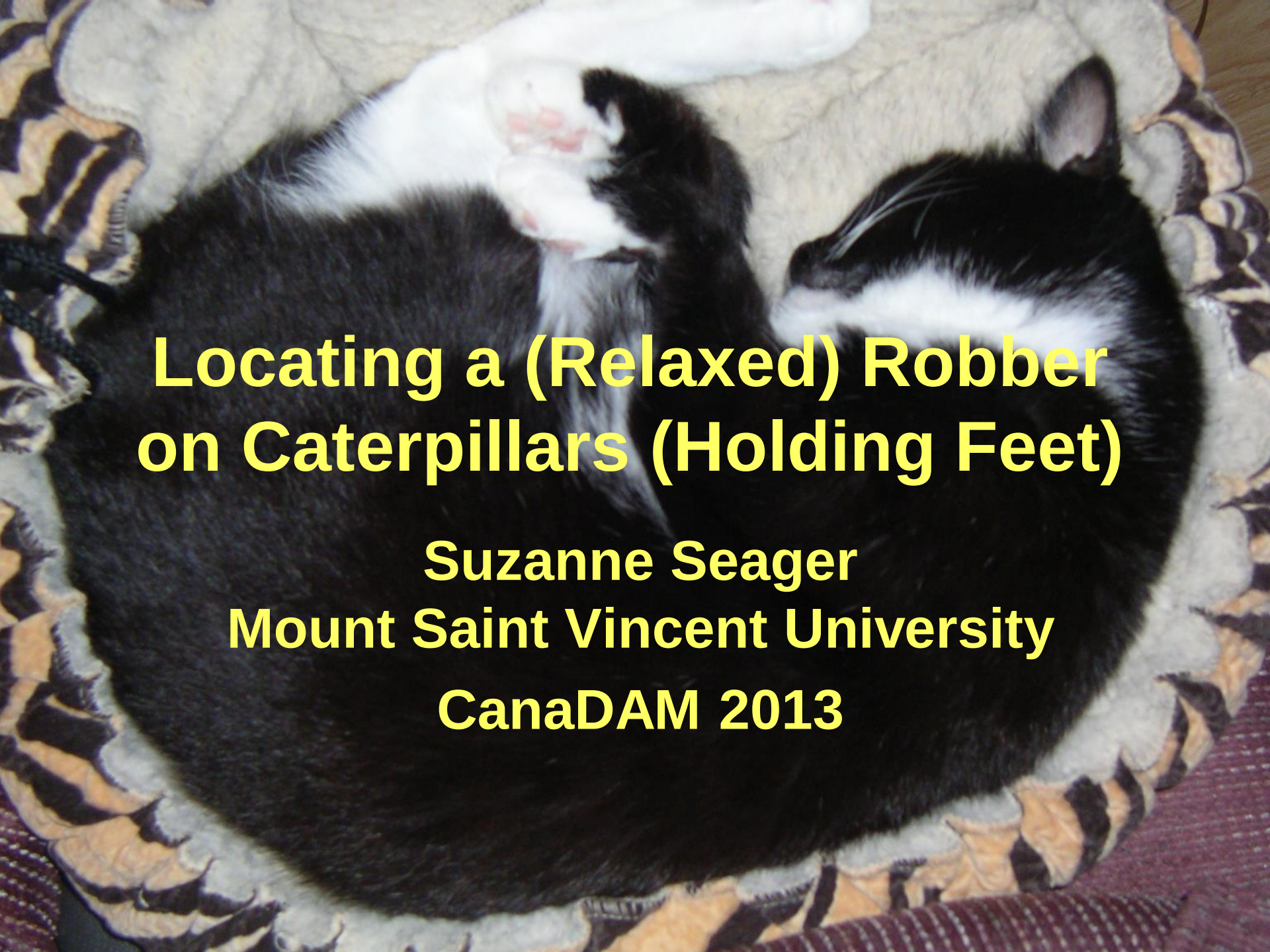


Locating a (Relaxed) Robber on Caterpillars (Holding Feet)

Suzanne Seager
Mount Saint Vincent University

CanaDAM 2013

Locating a Relaxed Robber on ~~a~~ Caterpillars Holding Feet

Consider the following game of a cop locating a robber on a connected graph. At each turn, the cop chooses a vertex of the graph to probe and receives the distance from the probe to the robber. If she can uniquely locate the robber after this probe, then she wins. Otherwise the robber may stay put or move to any vertex adjacent to his location ~~other than the probe~~. The cop's goal is to minimize the number of probes required to locate the robber; the robber's goal is to avoid being located. We consider an optimal cop strategy for caterpillars holding feet.

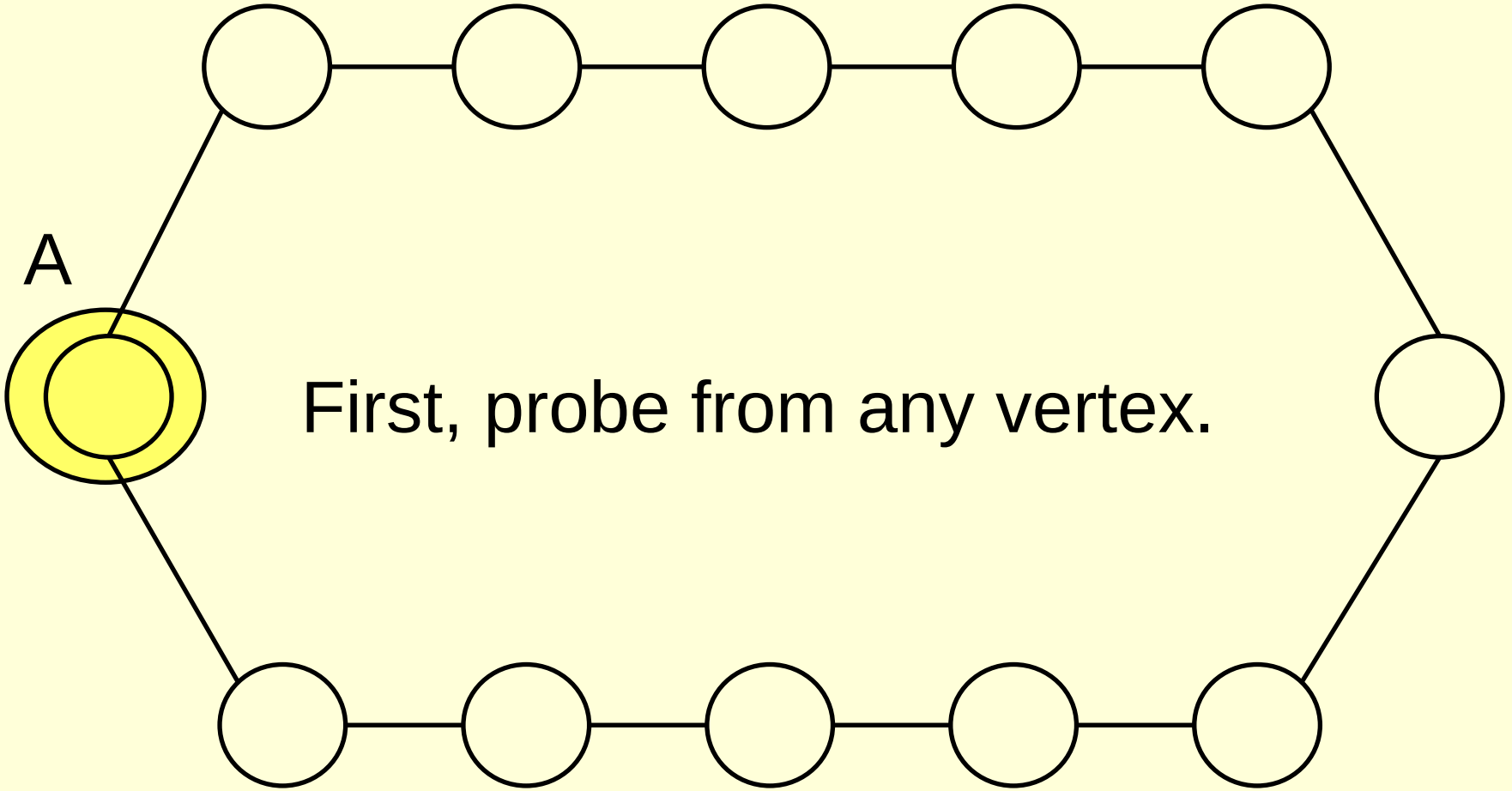
History

- http://jeux.lulu.pagesperso-orange.fr/
- Robber (Minou) cannot move
- Graph location [Slater/Harary-Melter 76]
- Cop/Robber [Quillot/Nowakowski-Winkler]
- Why can’t Robber move to last probe? [Dawson/Carraher-Choi-Delcourt-Erickson-West 2012]

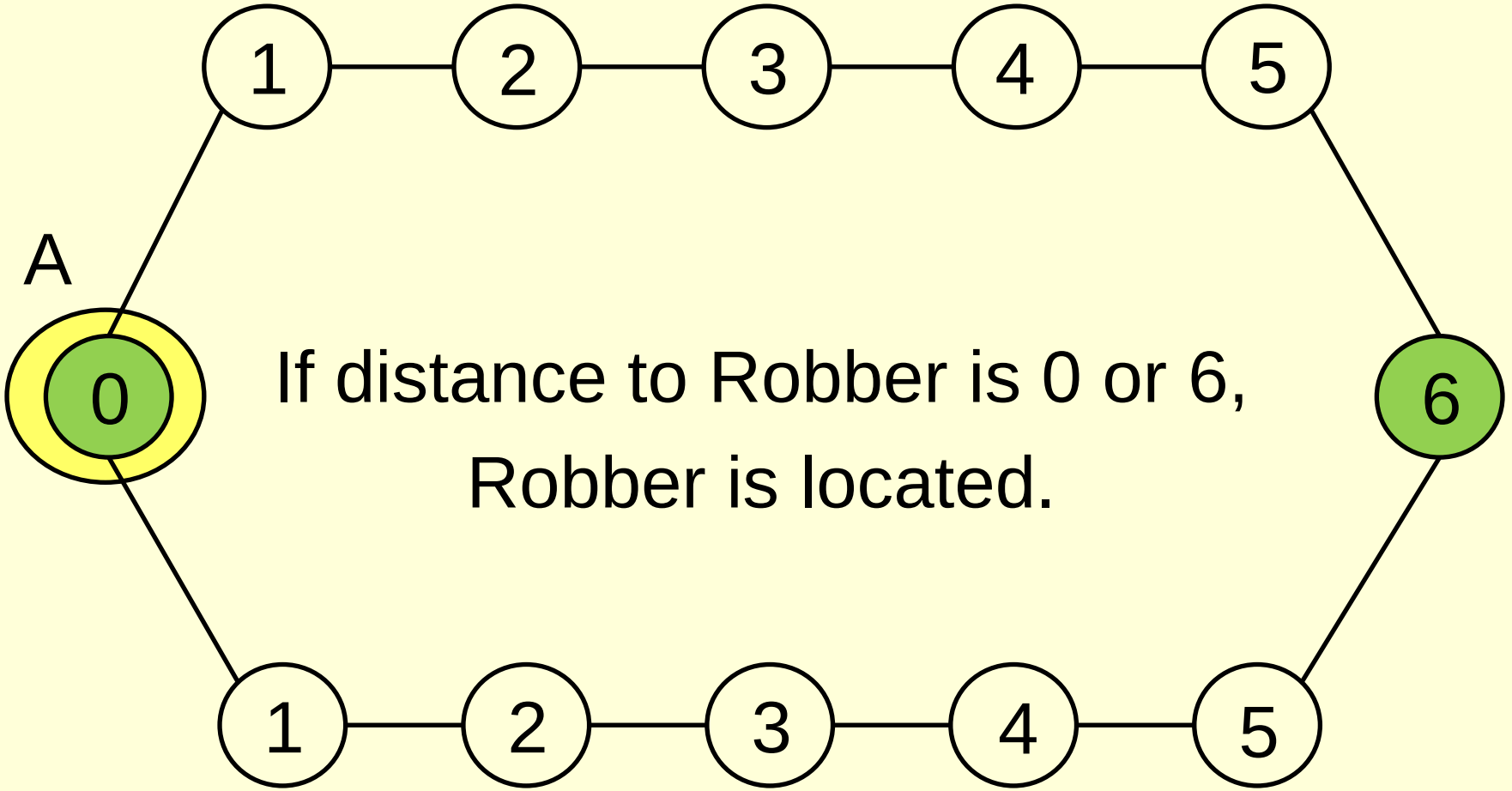
No-Backtrack Robber Locating Game on a Graph

1. Robber hides on a vertex.
2. Cop chooses a probe vertex, and sends a probe from that vertex to Robber, determining the distance between them.
3. Robber either stays put or moves to a neighbouring vertex ~~(but Robber NEVER moves to the probe vertex)~~.
4. Steps 2 and 3 are repeated until Cop wins by locating Robber (this may never happen, in which case Robber wins).

Example: Strategy for 12-Cycle

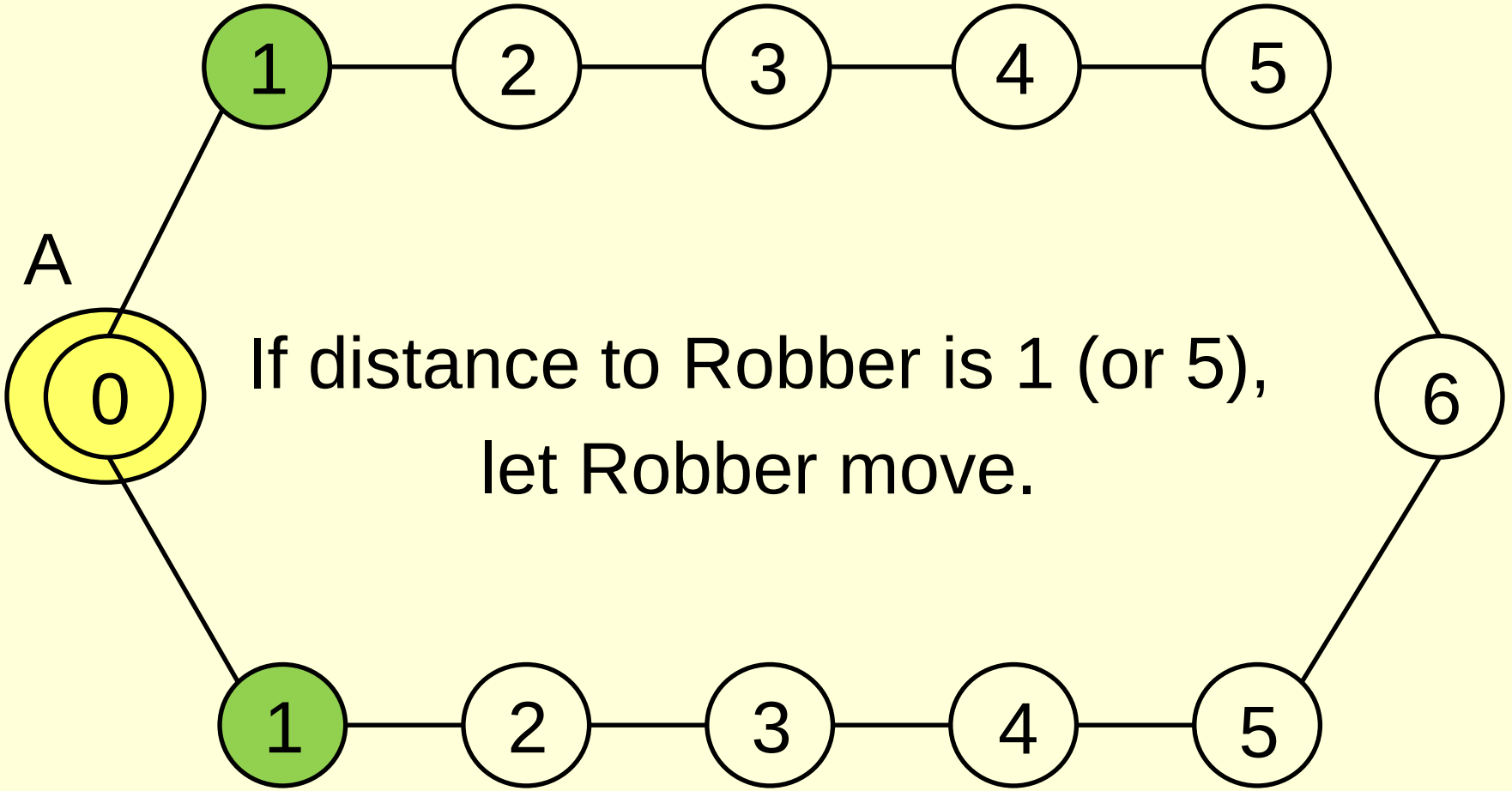


Example: Strategy for 12-Cycle

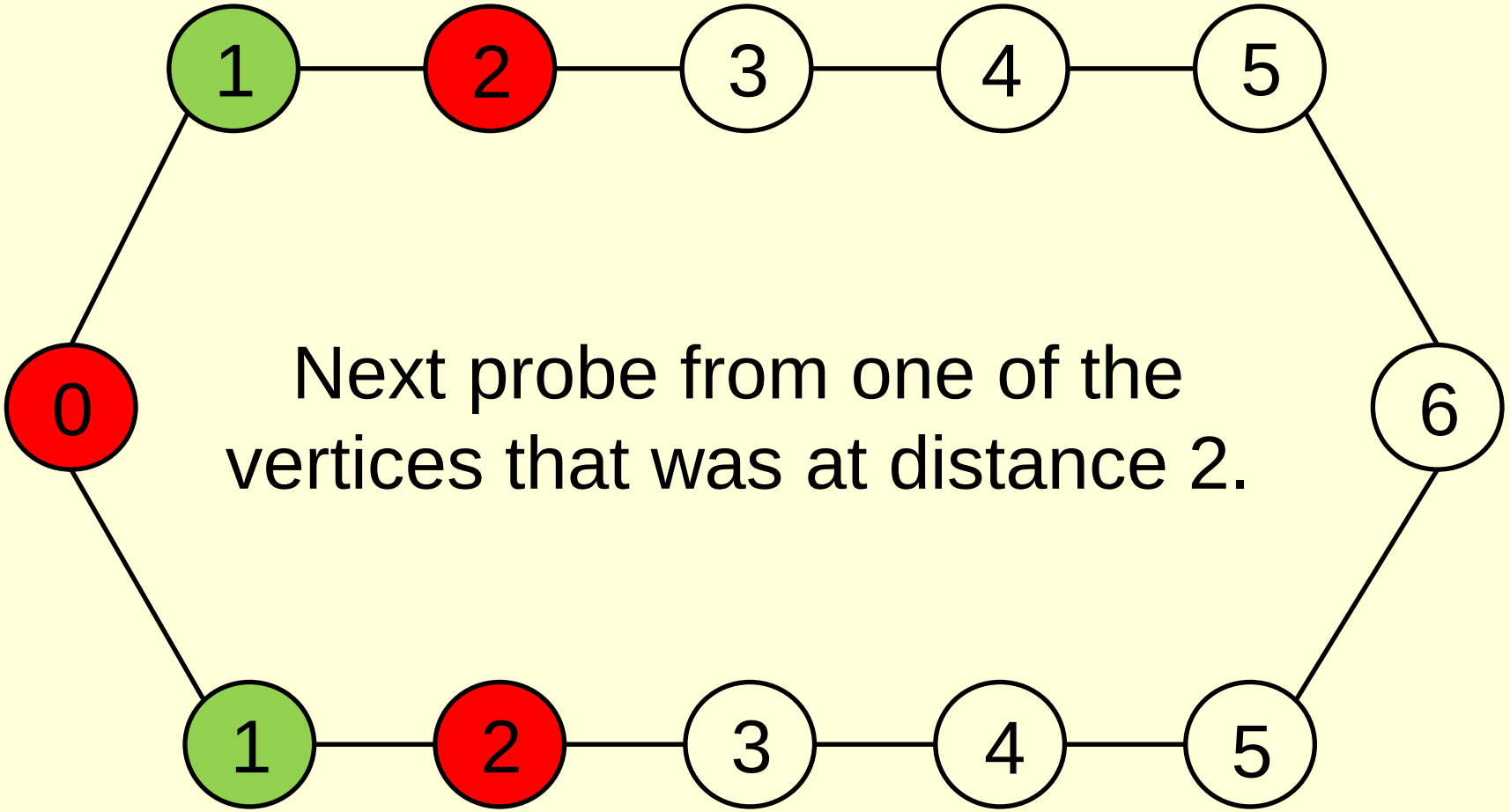


A0 or A6

Example: Strategy for 12-Cycle

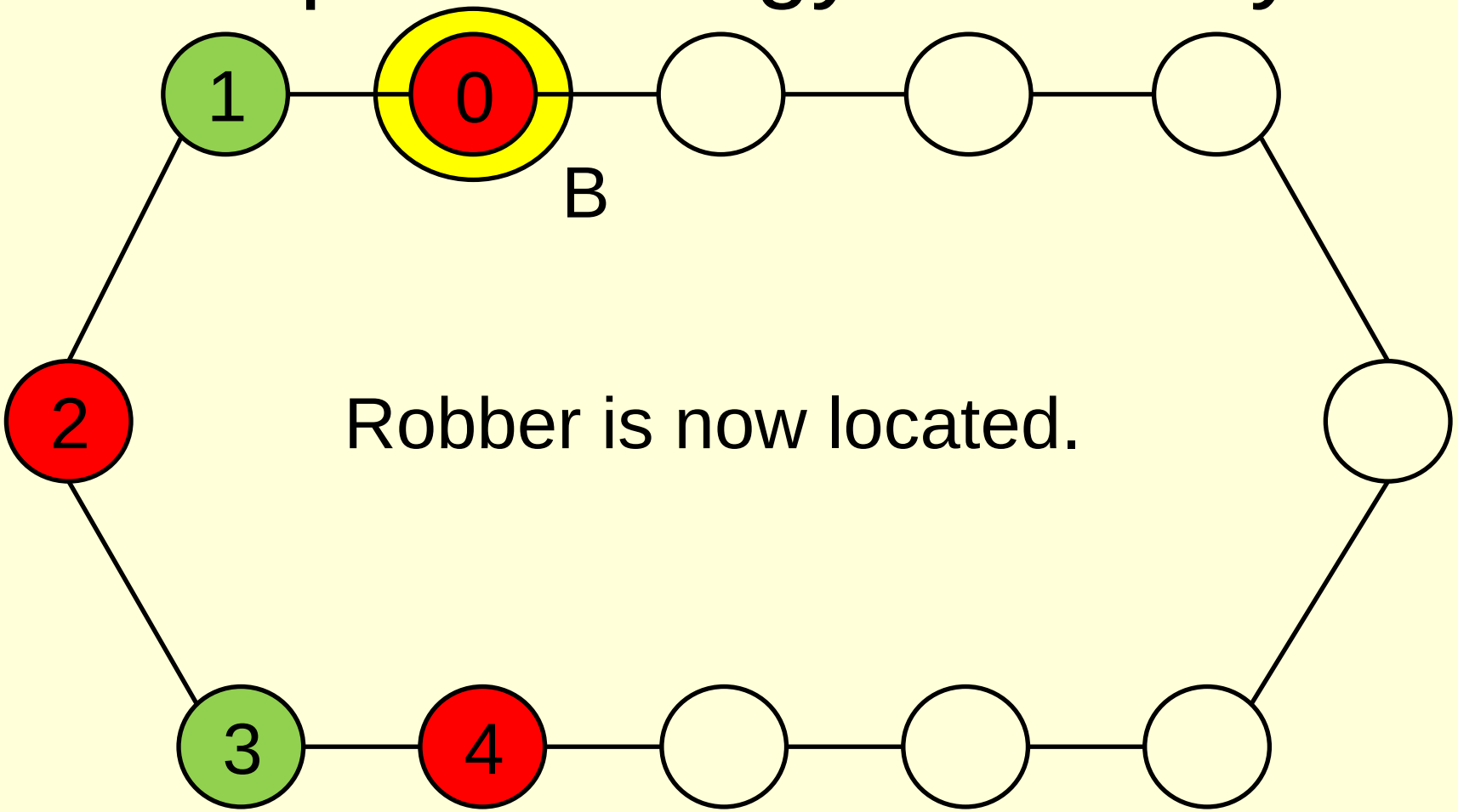


Example: Strategy for 12-Cycle

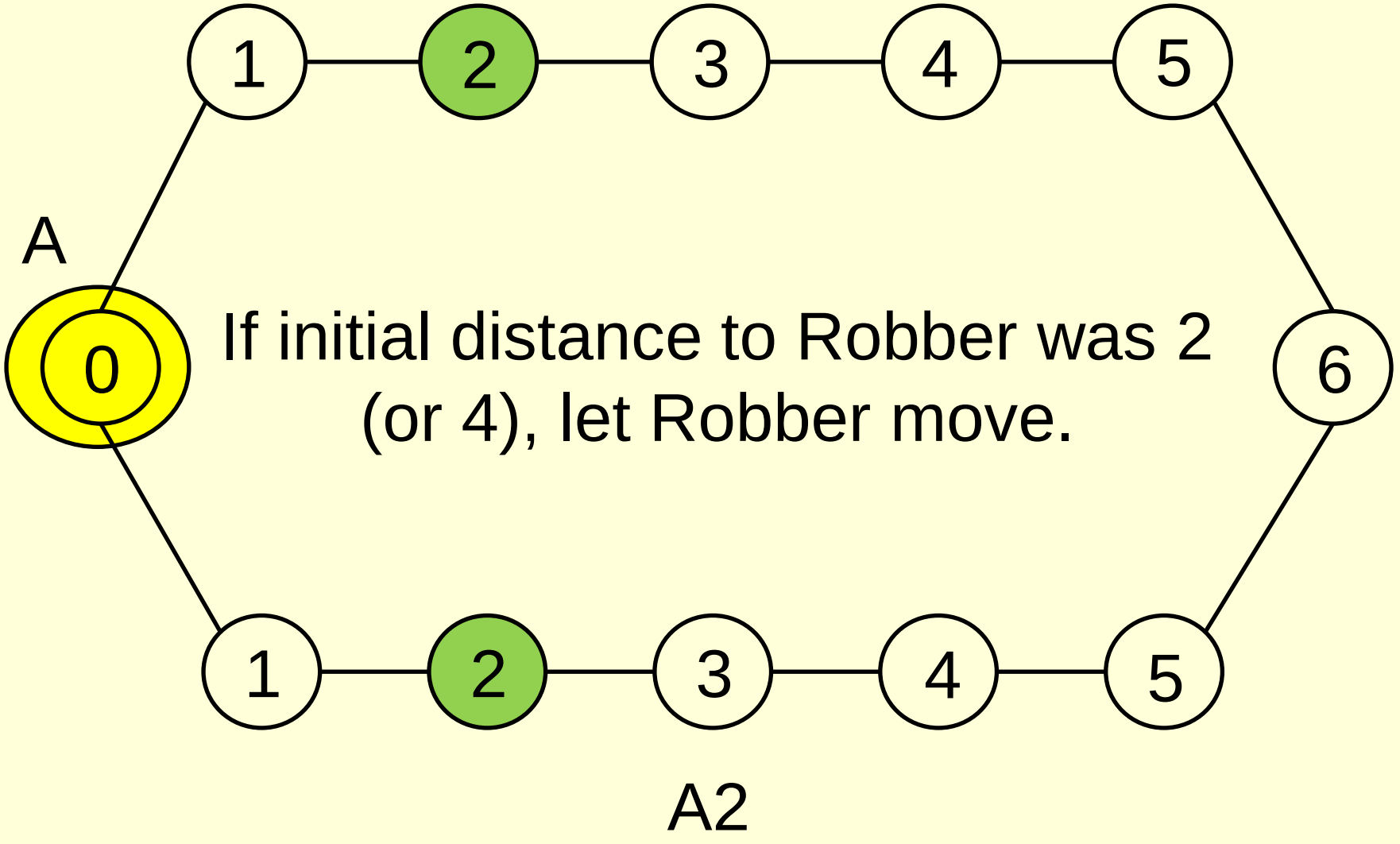


A1

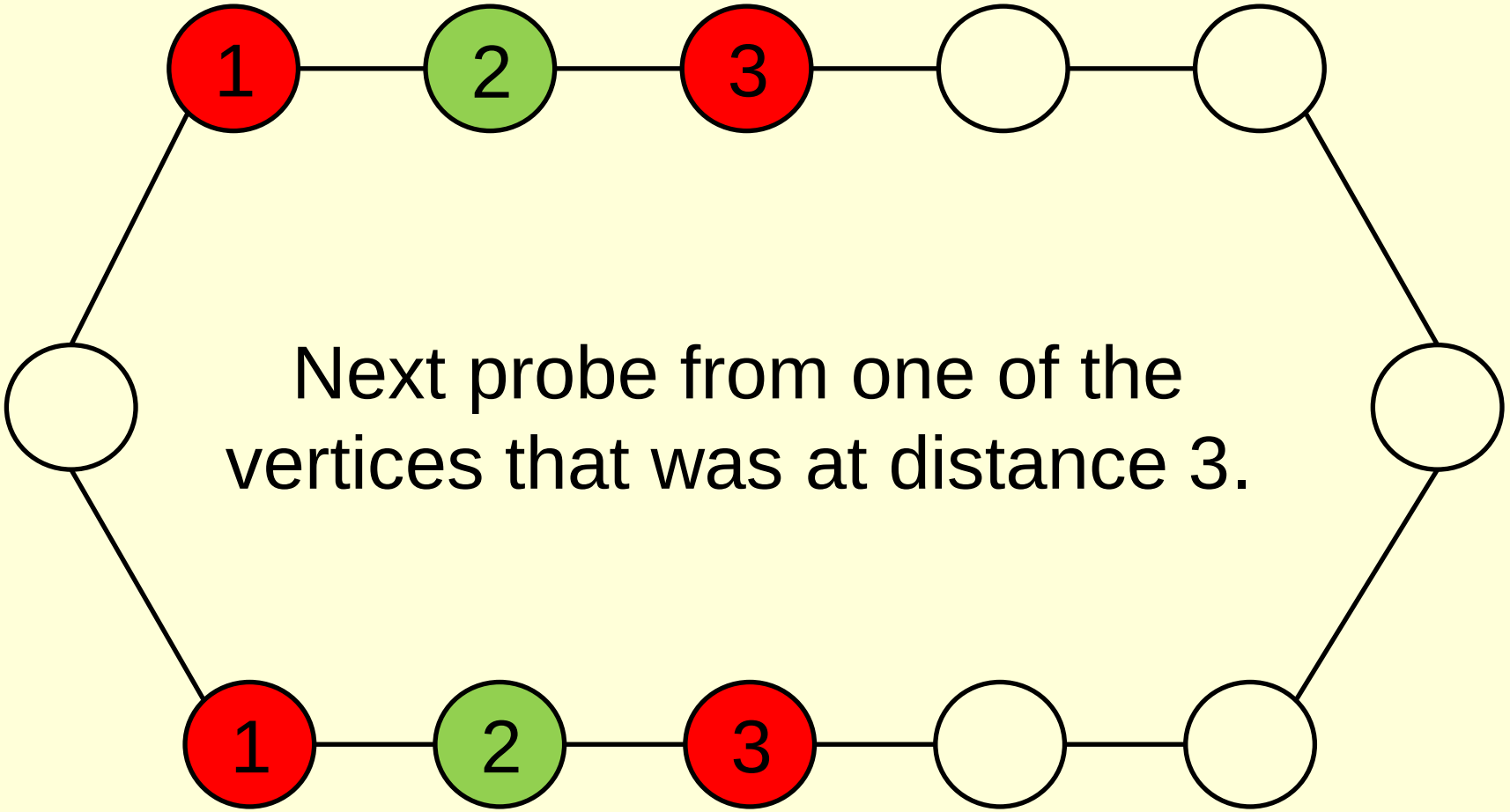
Example: Strategy for 12-Cycle



Example: Strategy for 12-Cycle

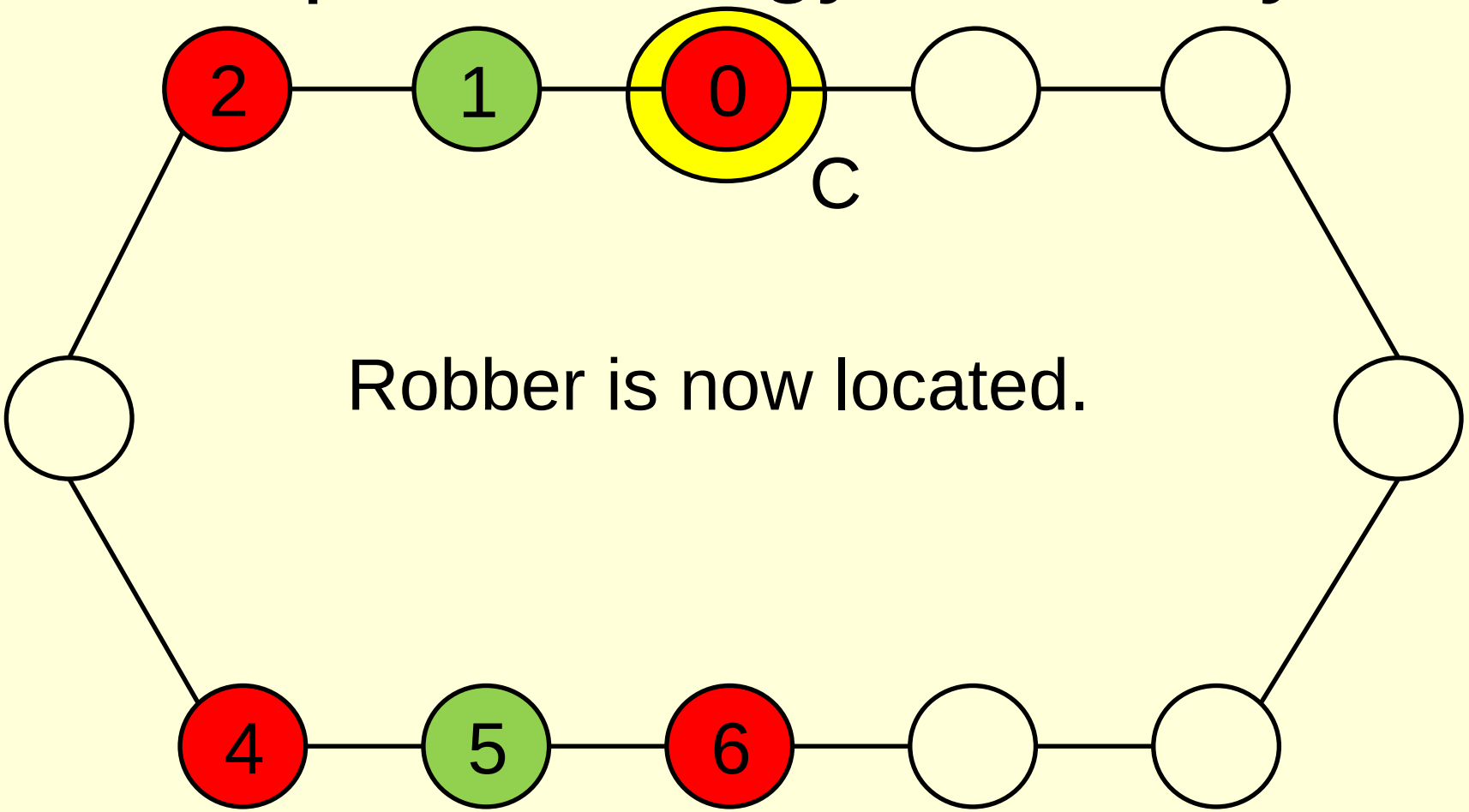


Example: Strategy for 12-Cycle



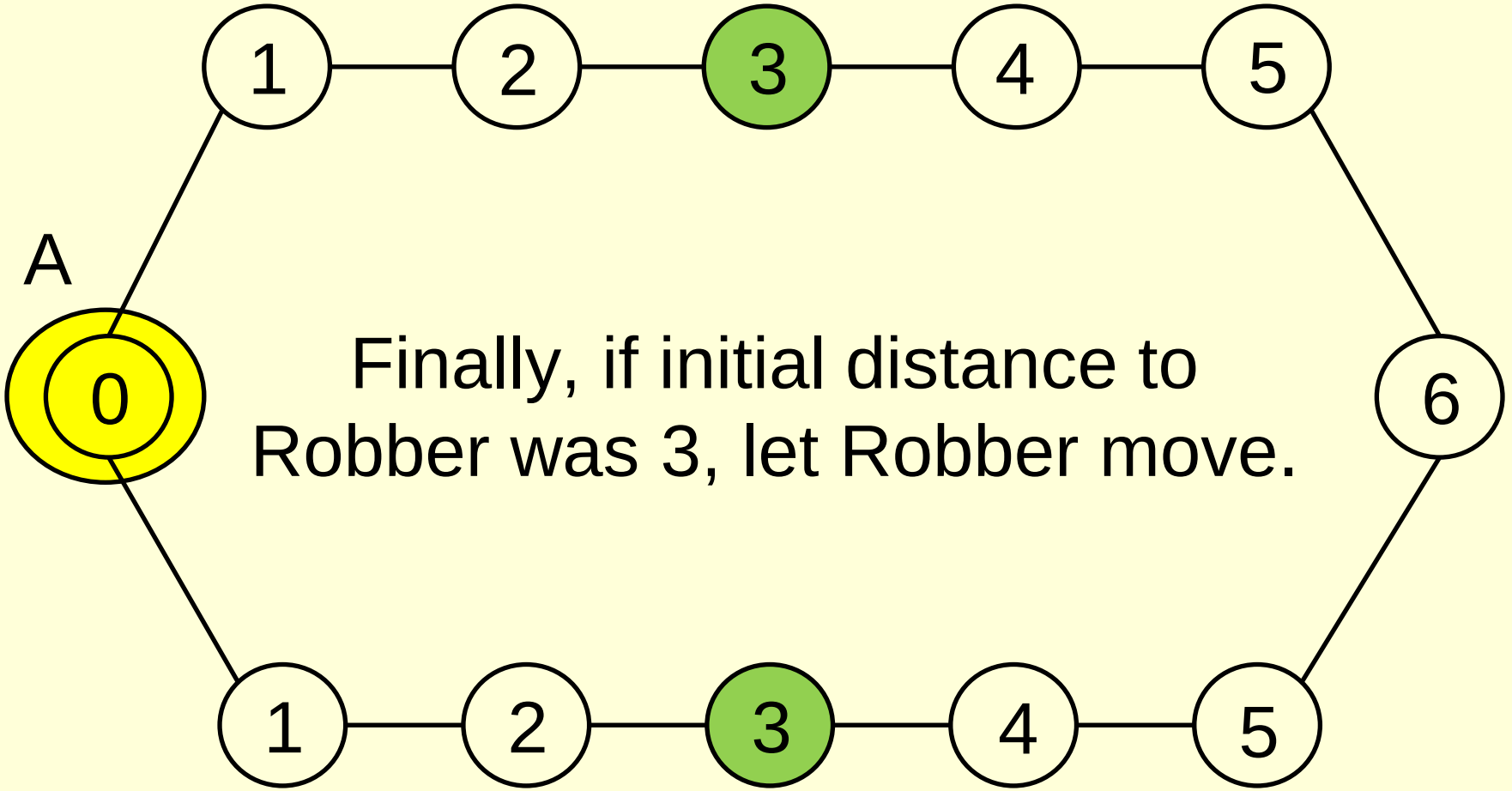
A2

Example: Strategy for 12-Cycle

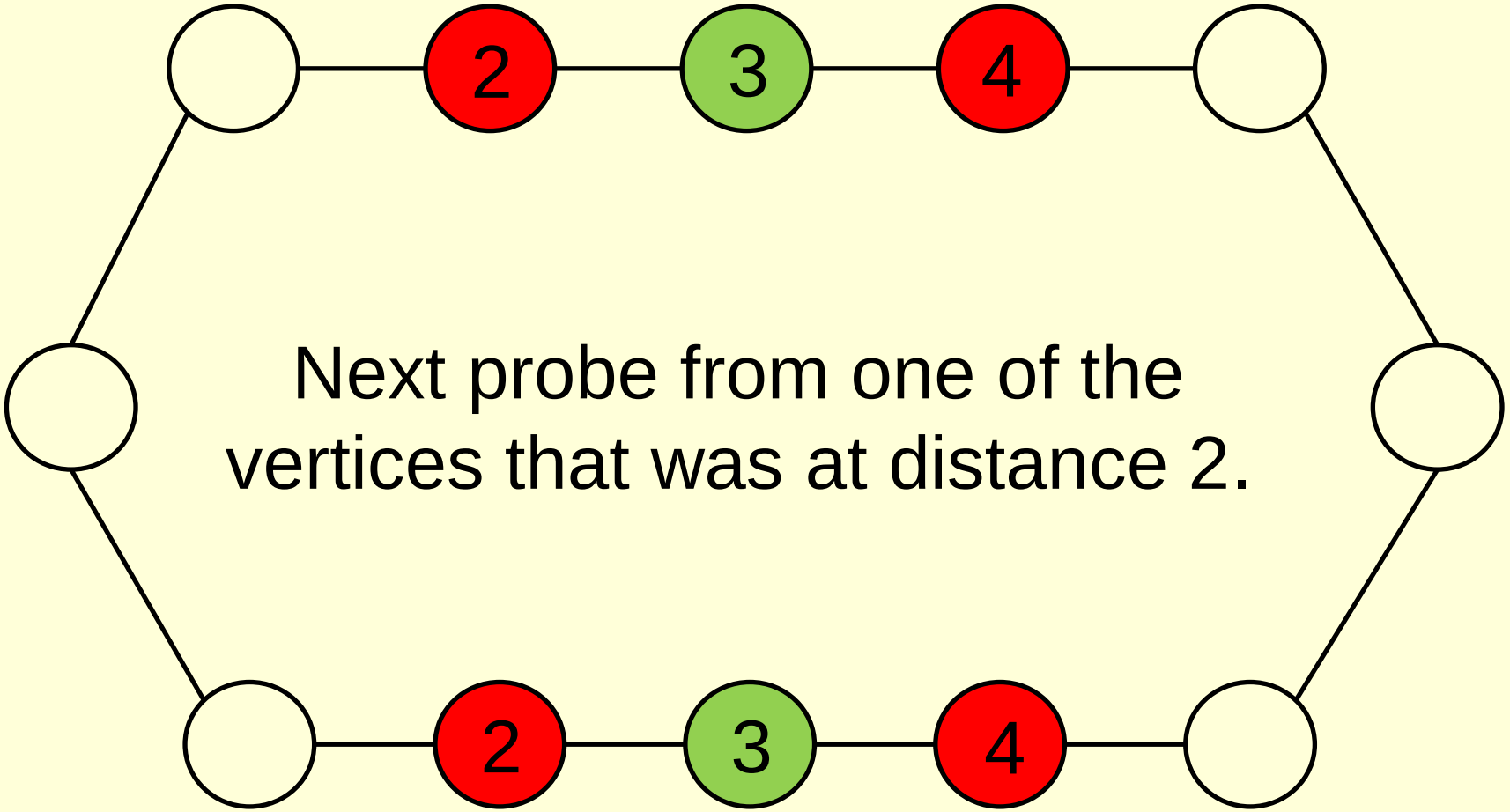


A1C

Example: Strategy for 12-Cycle

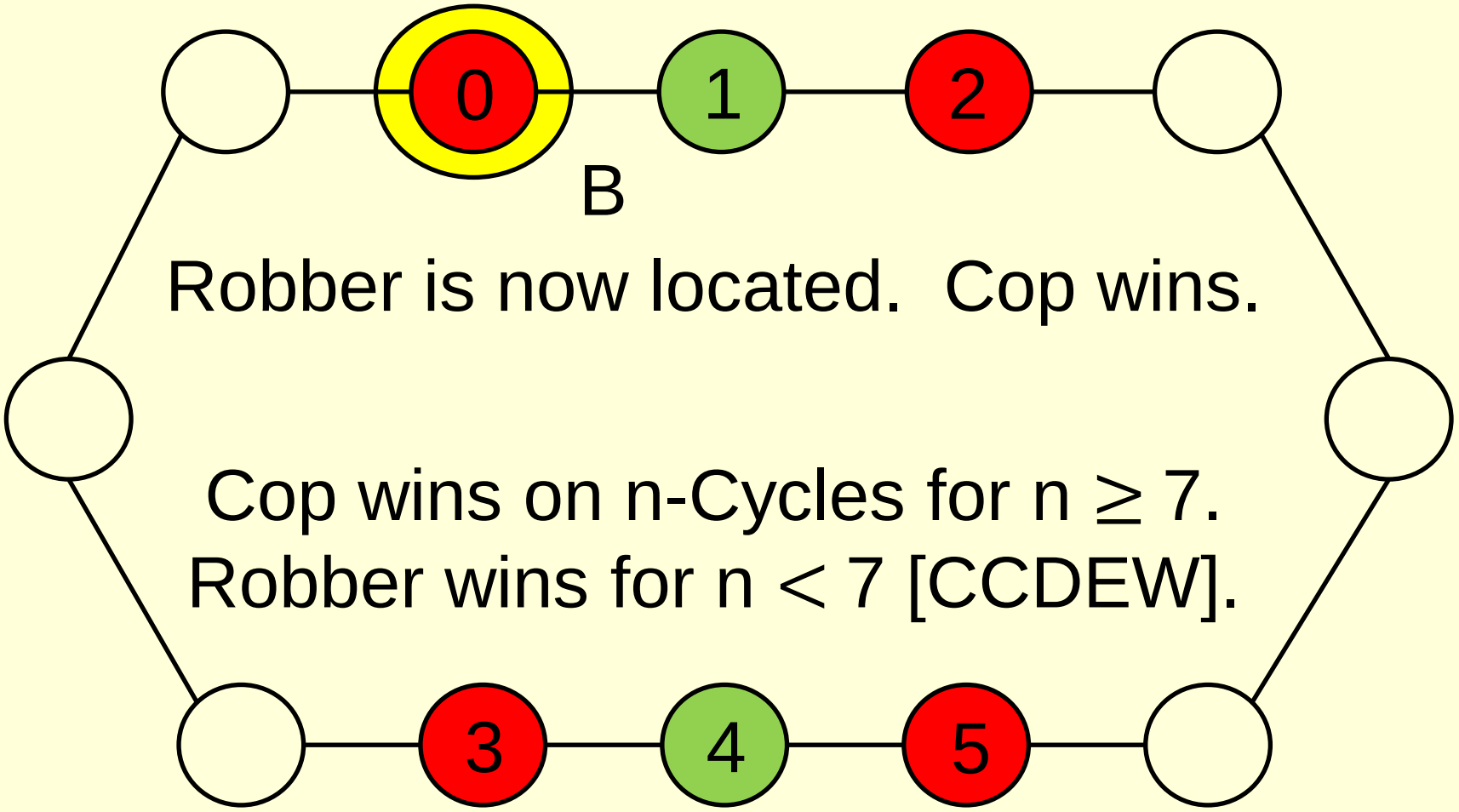


Example: Strategy for 12-Cycle



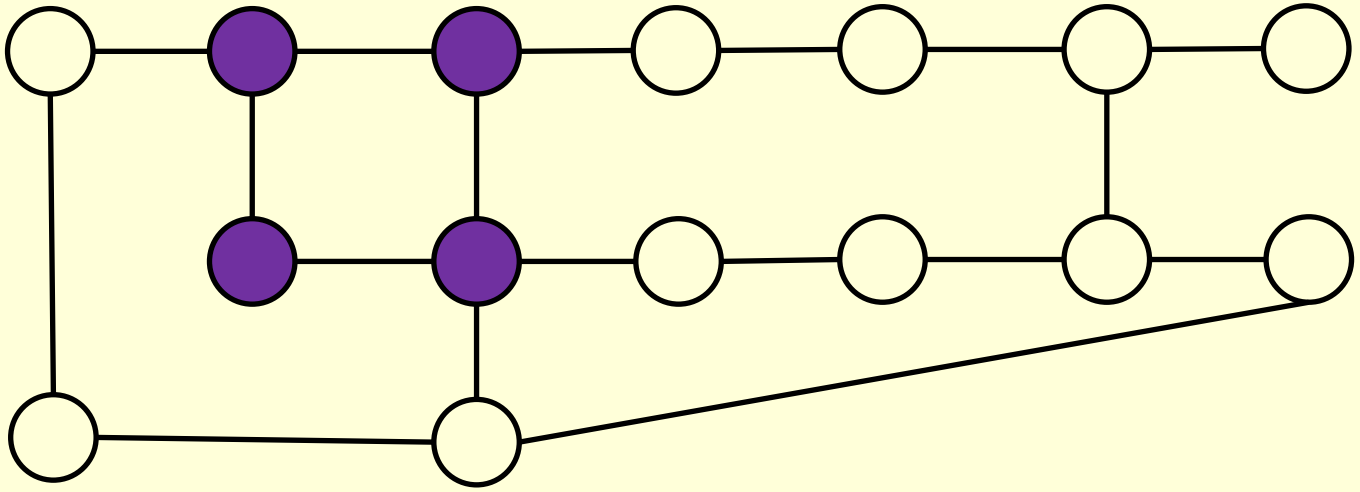
A3

Example: Strategy for 12-Cycle



Small Cycles are “Hideouts”

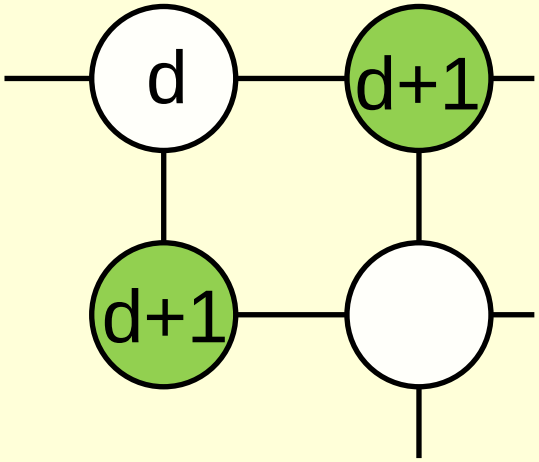
Theorem [Carraher, Choi, Delcourt, Erickson, & West, 2012] On any graph with a cycle C of length 5 or less, Robber can win without ever leaving C.

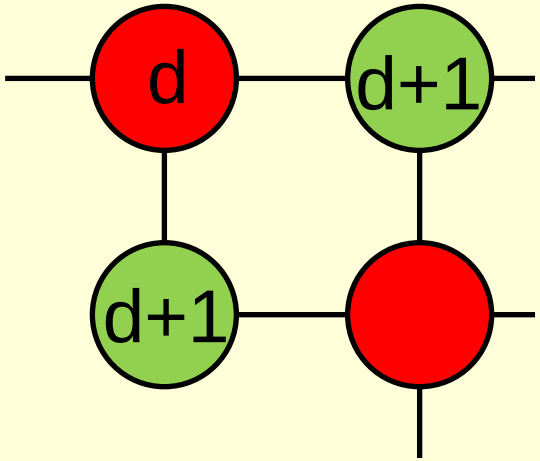


Why is a 4-cycle a hide-out?

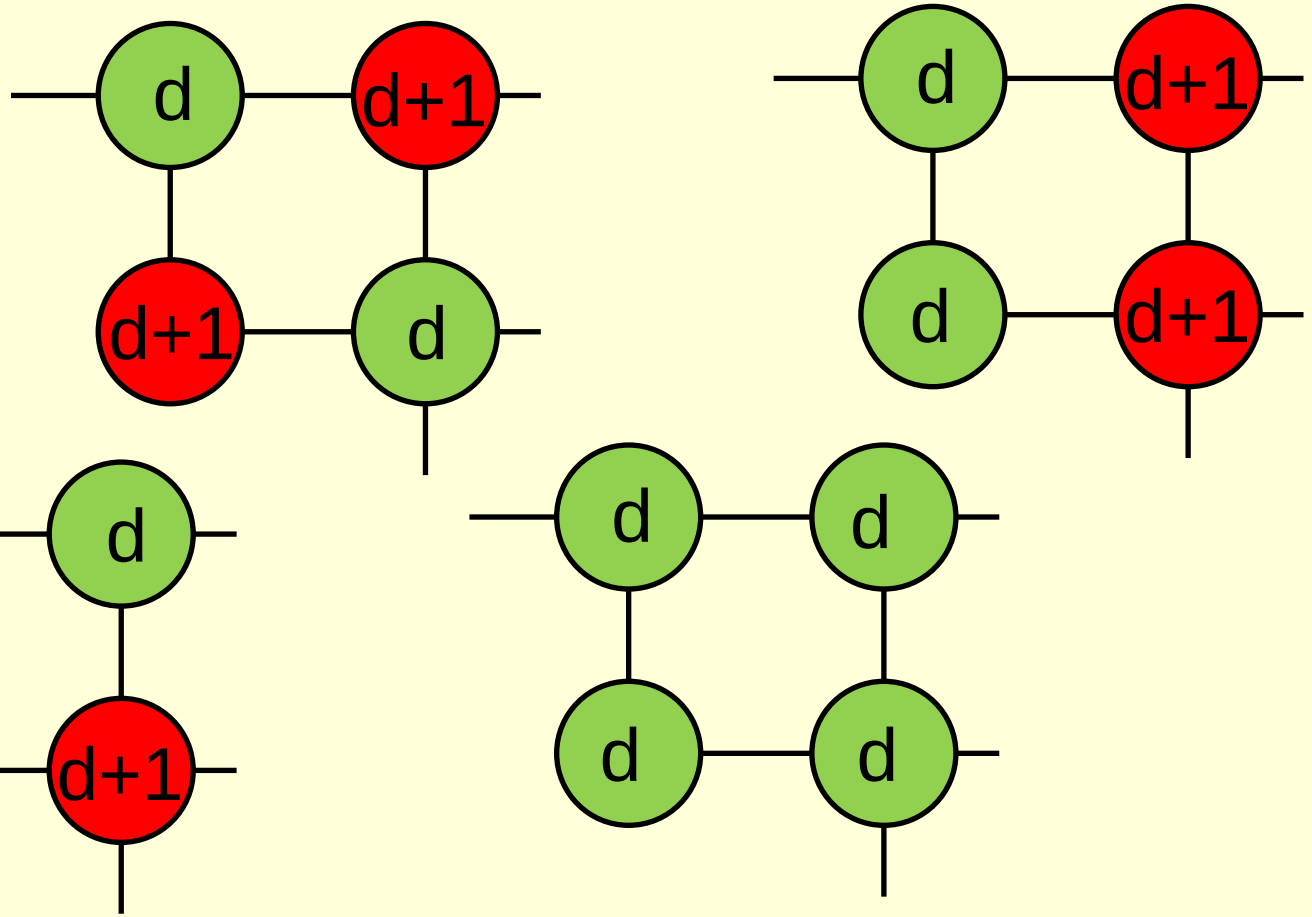
Count how many vertices on the 4-cycle are at **minimum distance d from the first probe**.

If **1**, its two neighbours are at distance d+1, so Robber is not located and, after moving, can be anywhere on the 4-cycle.



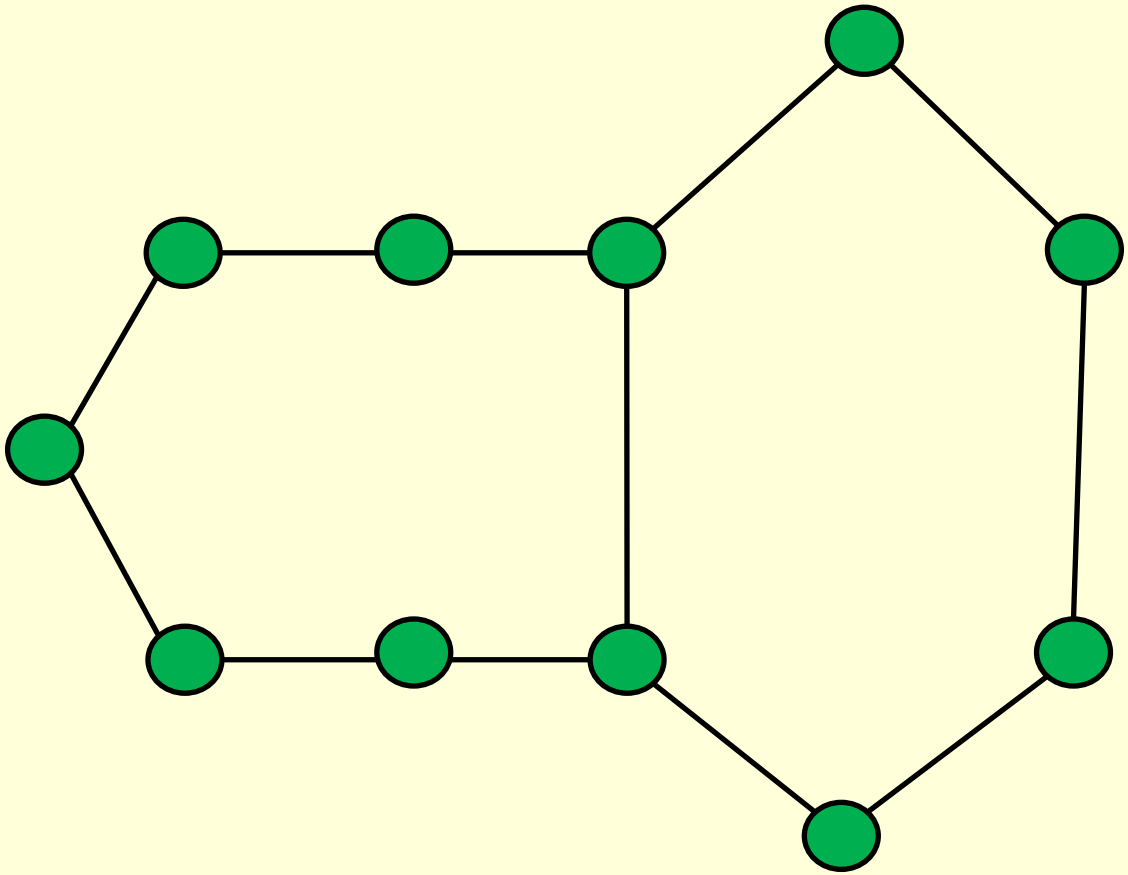


If the number of vertices at minimum distance d from the first probe is **2, 3 or 4**, Robber is not located and, after moving, can be anywhere on the 4-cycle.



A 6-cycle need not be a hideout

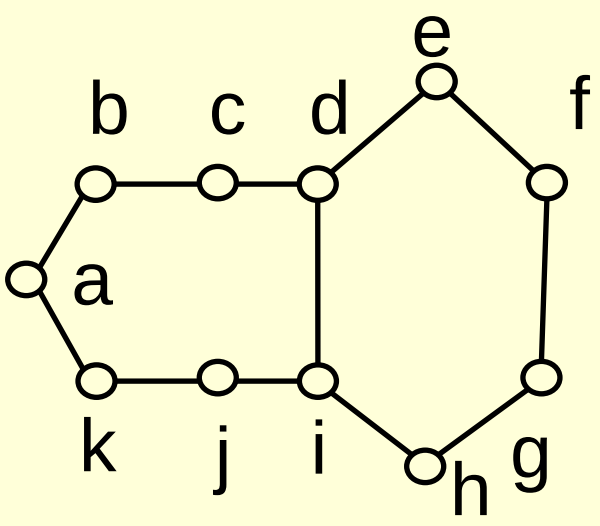
Example: for the graph below, Cop wins.

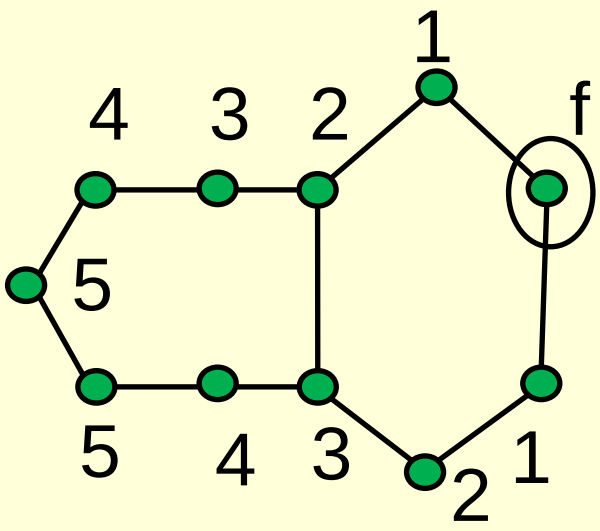


Strategy for Cop

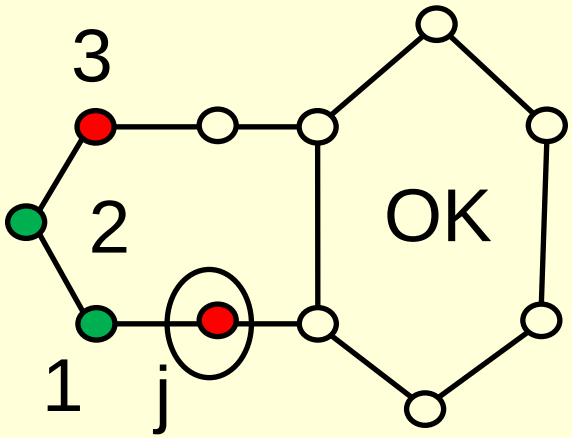
A: probe f, then d=1 to **B**, d=2 to **C**, d=3 to **D**, d=4 to **E**, d=5 to **F**.

F: probe j, then d= 1,2,3 locates robber.





A f



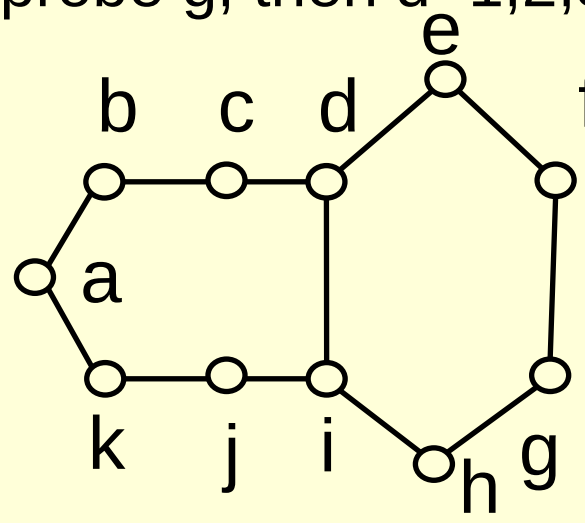
F f5j

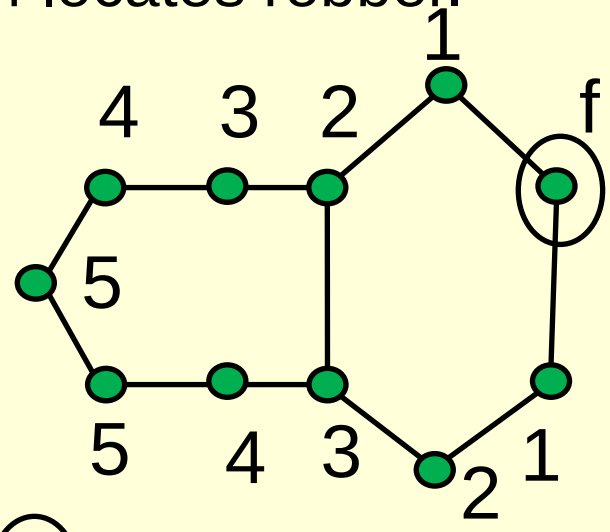
A: probe f, then d=1 to **B**, d=2 to **C**, d=3 to **D**, d=4 to **E**, d=5 to **F**.

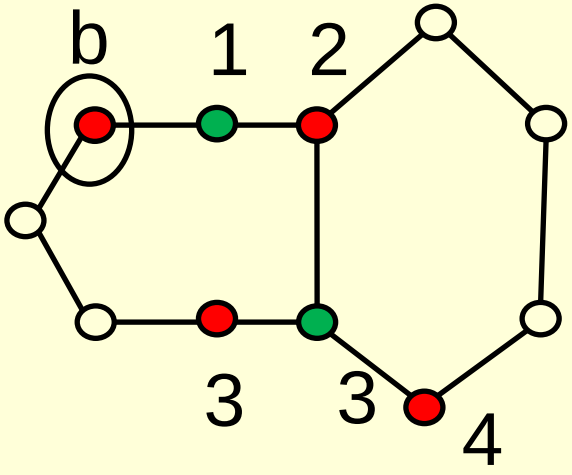
D: probe b, then d=1,2,4 locates robber, d=3 to **T**.

T: probe e, then d=1,2,4 locates robber, d=3 to **K'**.

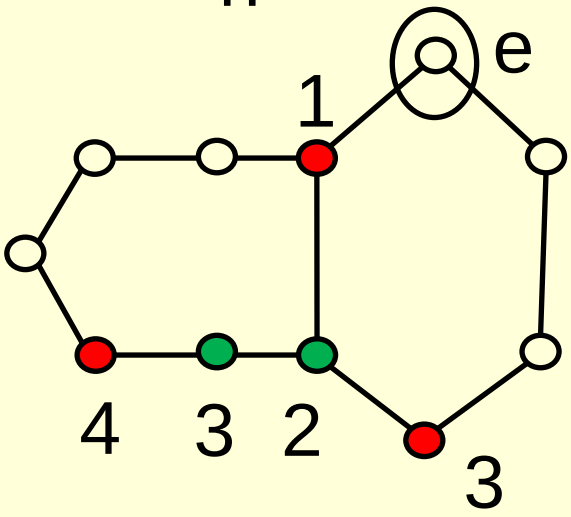
K': probe g, then d=1,2,3,4 locates robber.



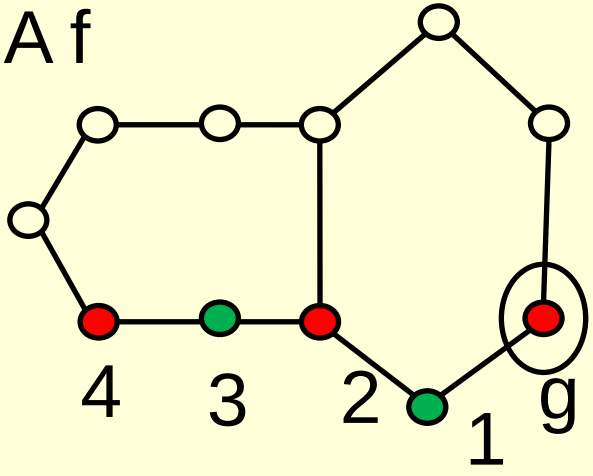




D f3b

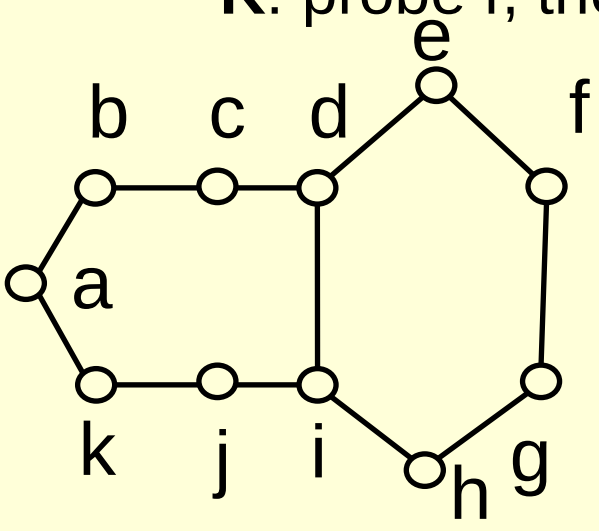


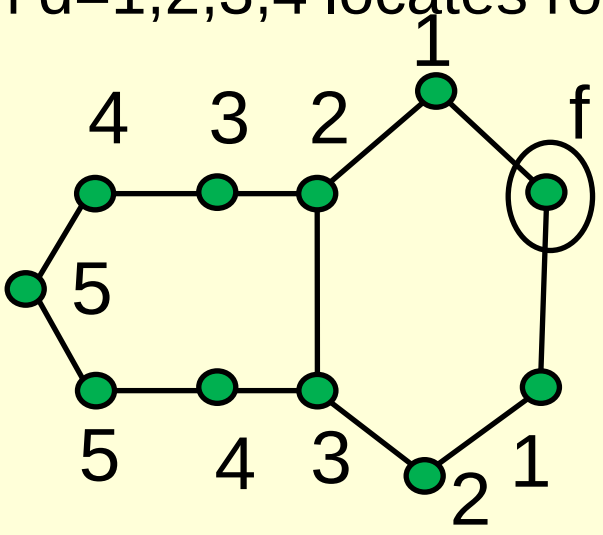
T f3b3e



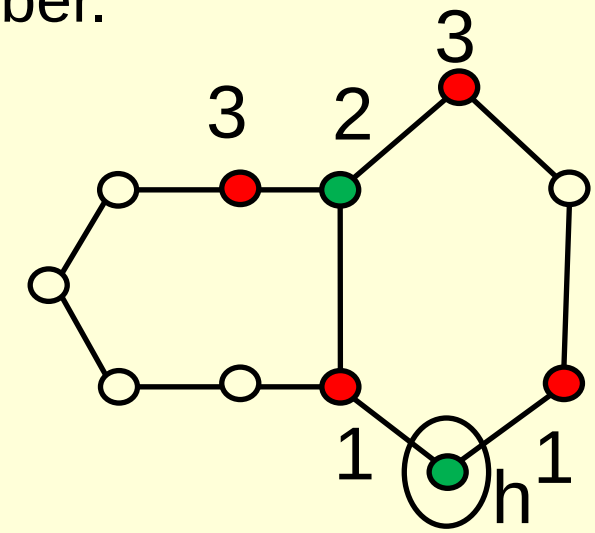
K' f3b3e2g

A: probe f, then d=1 to **B**, d=2 to **C**, d=3 to **D**, d=4 to **E**, d=5 to **F**.
C: probe h, then d=1 to **Q**, d=3 to **K**, d=2 locates robber.
K: probe f, then d=1,2,3,4 locates robber.

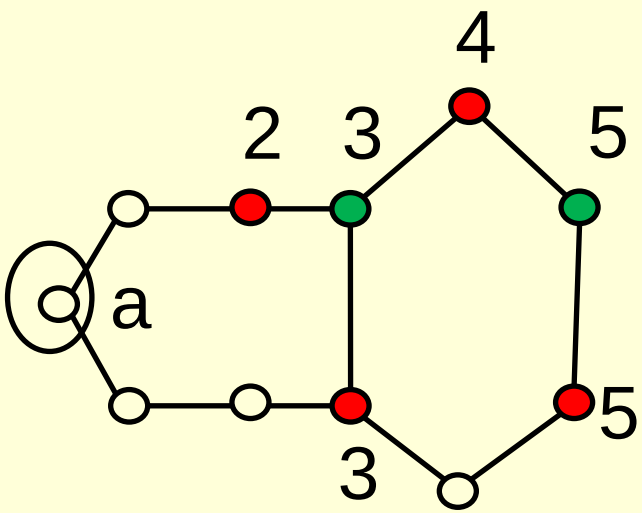




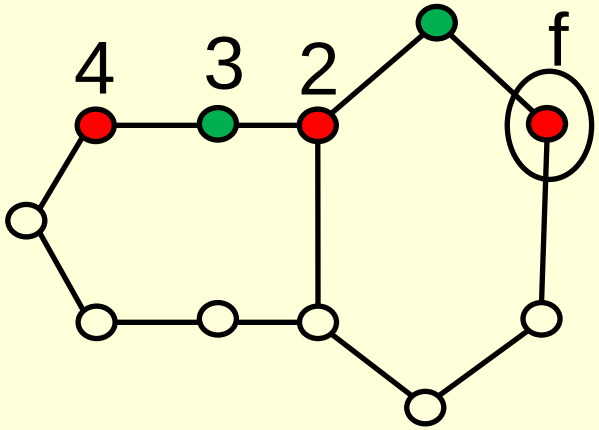
A f



C f2h



Q f1e1a

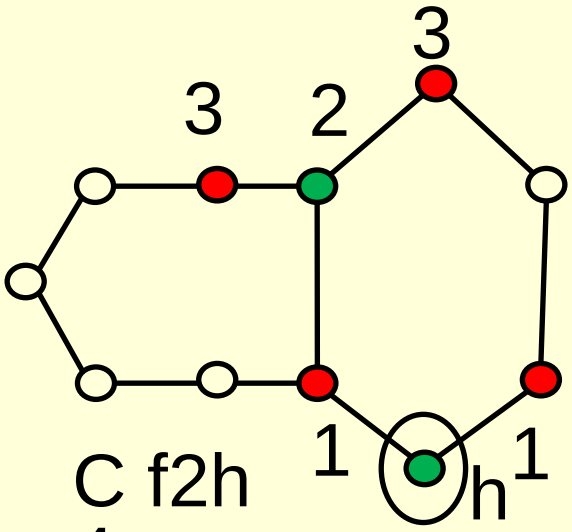


K f4b1k3h

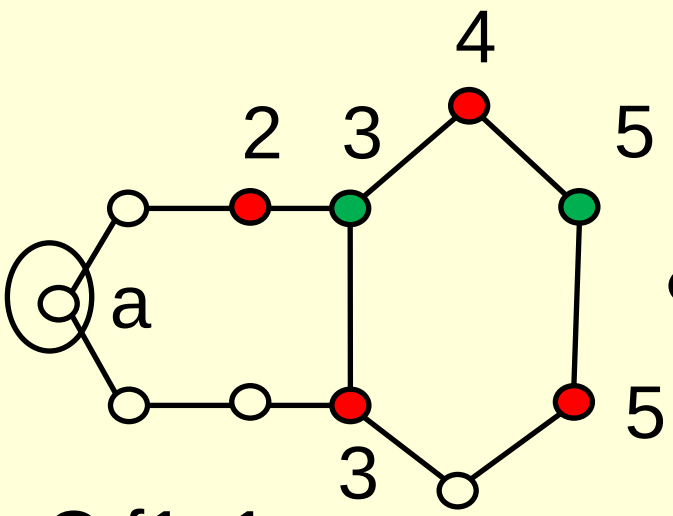
C: probe h, then d=1 to **Q…**

Q: probe a, then d=2,4 locates robber, d=3 to **N**, d=5 to **P**..

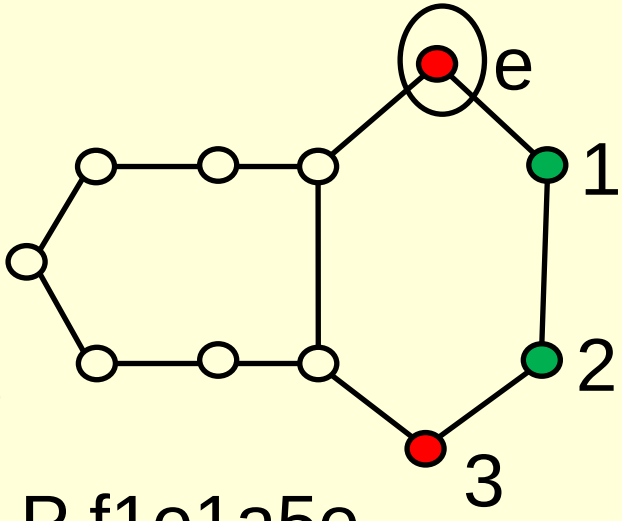
P: probe e, then d=1,2,3 locates robber.



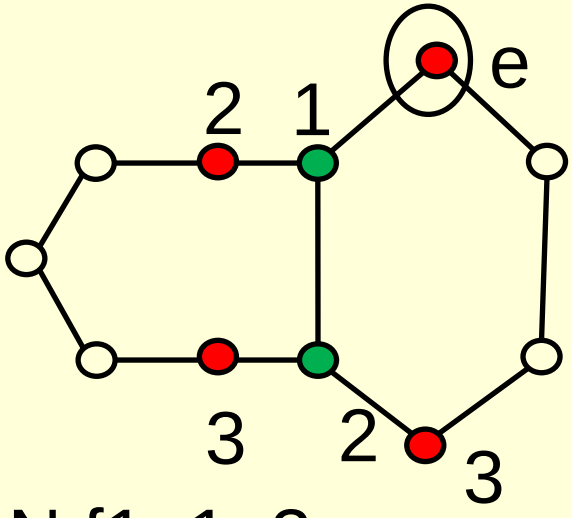
C f2h



Q f1e1a



P f1e1a5e

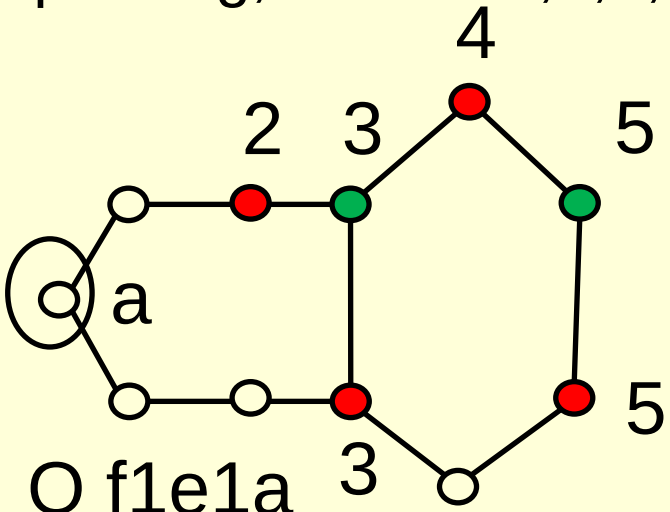


N f1e1a3e

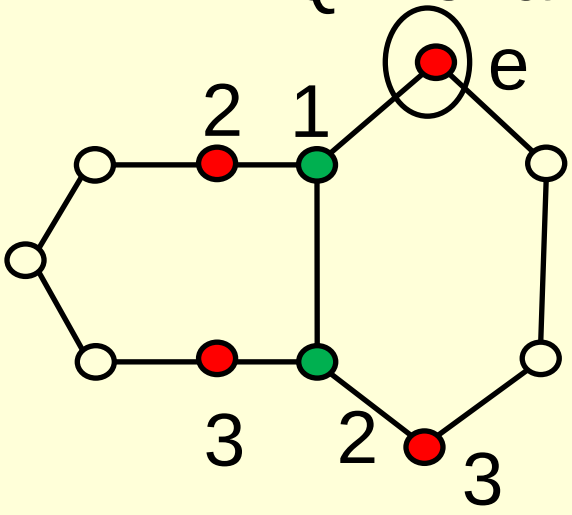
Q: probe a, then d=3 is **N**....

N: probe e, then d=1 locates robber, d=2 to **D** and d=3 to **K'**.

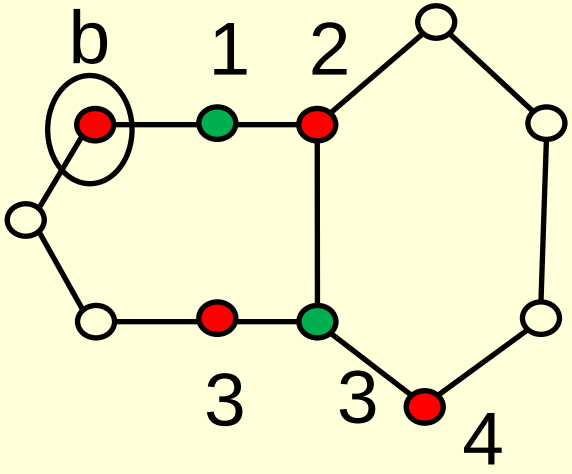
K': probe g, then d=1,2,3,4 locates robber.



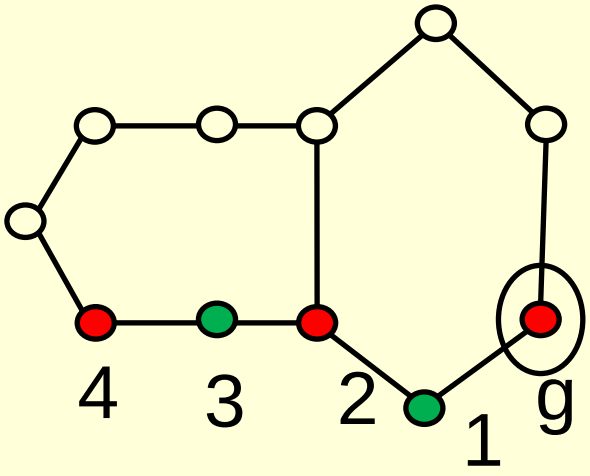
Q f1e1a



N f1e1a3e

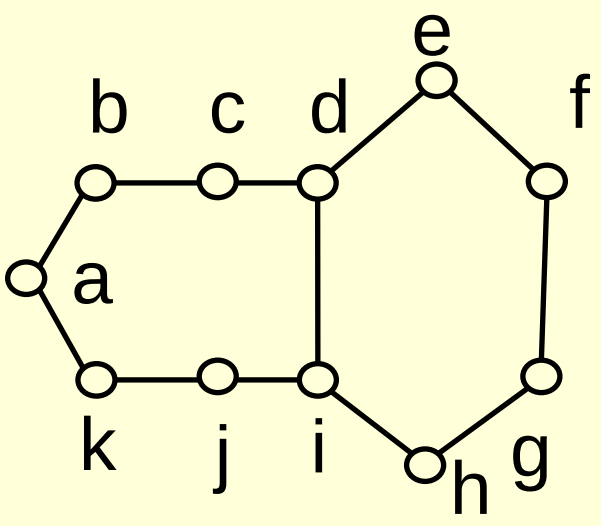


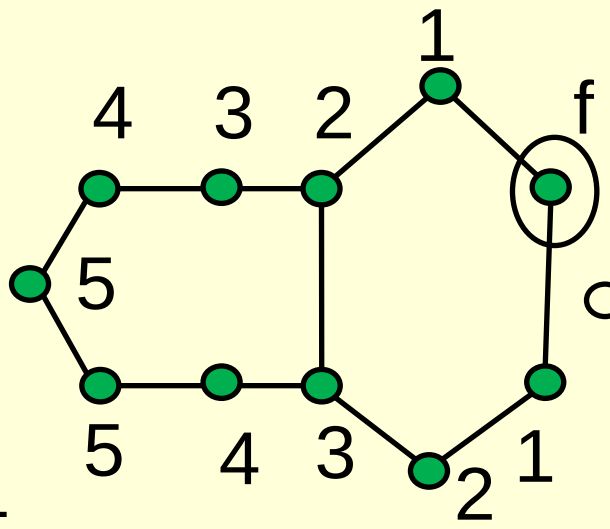
D f3b



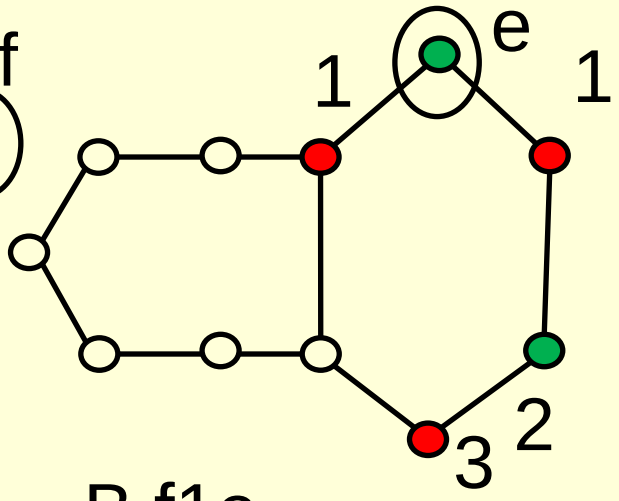
K' f3b3e2g

A: probe f, then d=1 to **B**, d=2 to **C**, d=3 to **D**, d=4 to **E**, d=5 to **F**.
B: probe e, then d=1 to **Q**, d=2,3 locates robber.

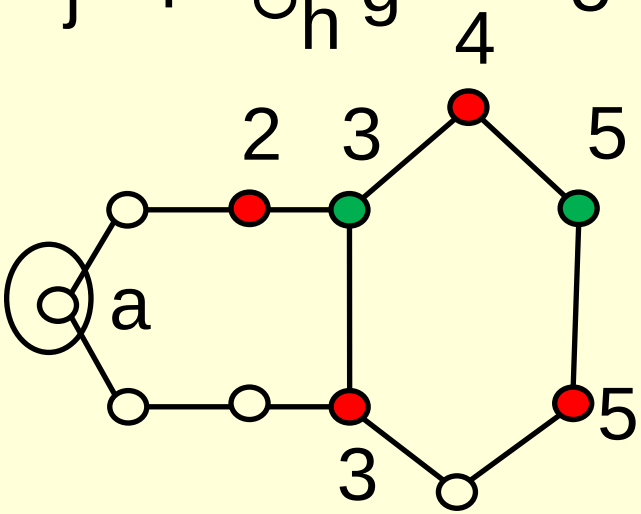




A f



B f1e

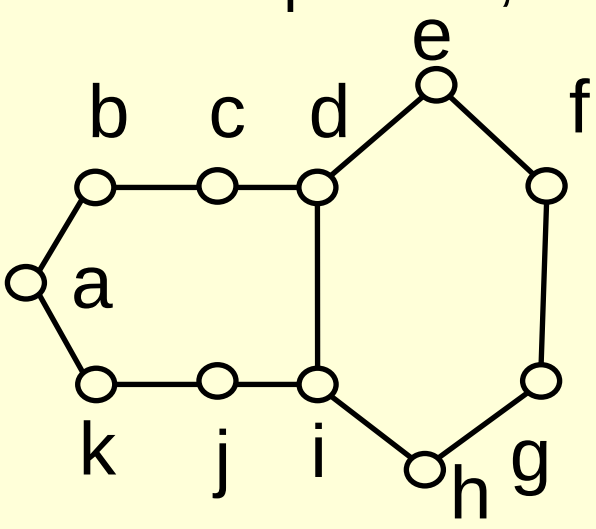


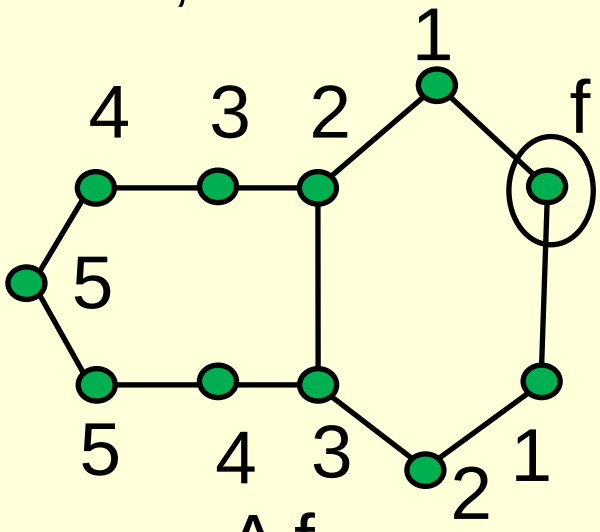
Q f1e1a

A: probe f, then d=1 to **B**, d=2 to **C**, d=3 to **D**, d=4 to **E**, d=5 to **F**.

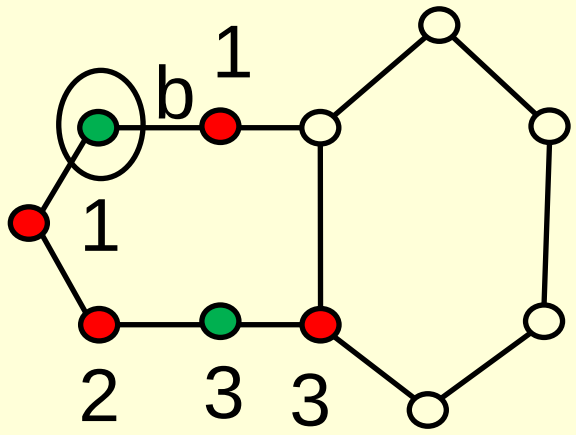
E: probe b, then d= 2 locates robber, d=1 to **G**, d=3 to **T**.

G: probe k, then d=1,2 locates robber, d=3 to **T'**.

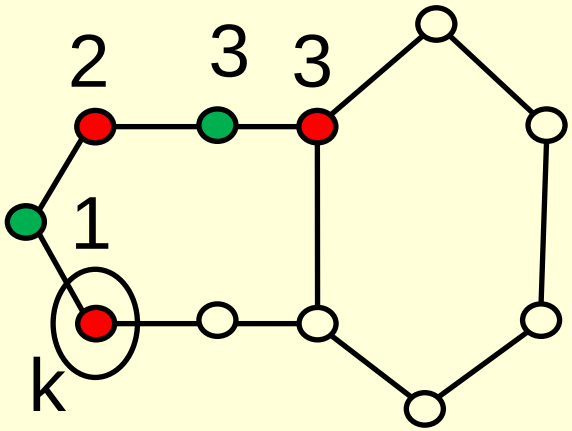




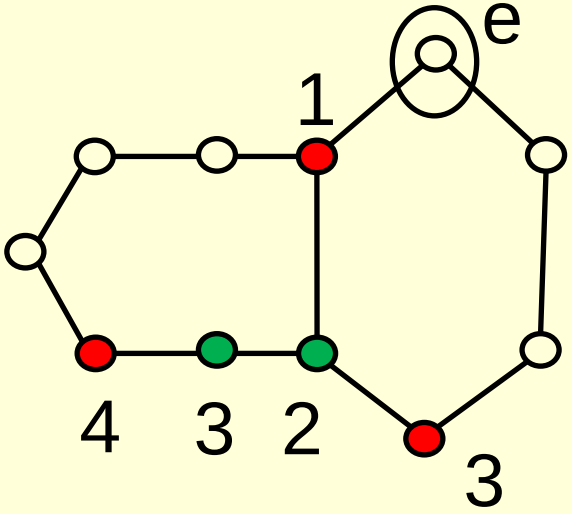
A f



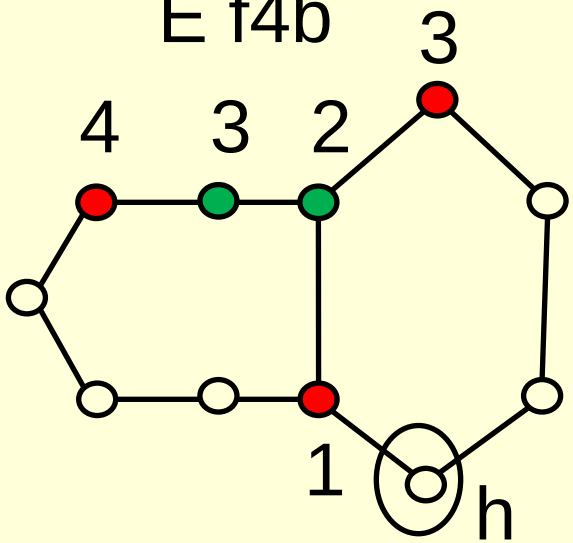
E f4b



G f4b1k



T f4b3e



T' f4b1k3h

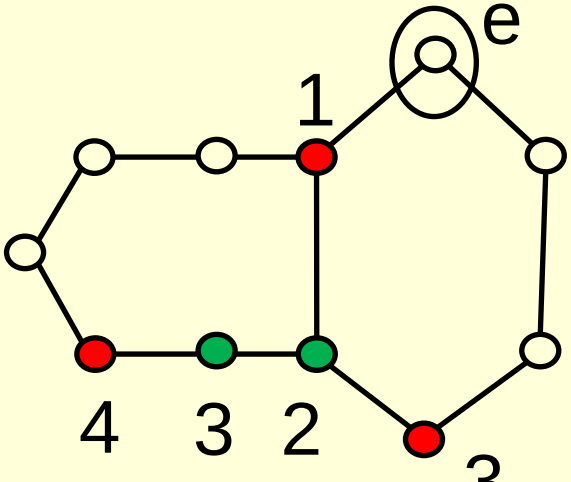
E: probe b, d=1 to **G**, d=3 to **T**…. **G**: probe k, d=3 to **T'**....

T: probe e, then d=1,2,4 locates robber, d=3 to **K'**.

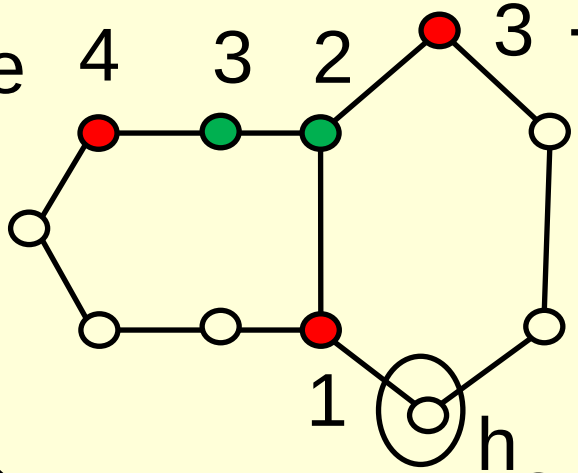
K': probe g, then d=1,2,3,4 locates robber

T': probe h, then d=1,2,4 locates robber, d=3 to **K**.

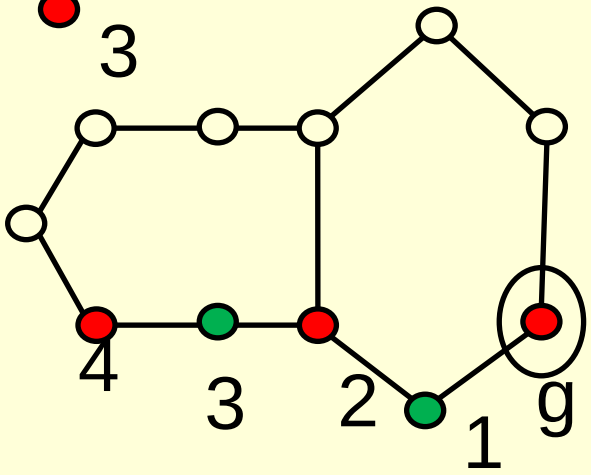
K: probe f, then d=1,2,3,4 locates robber.



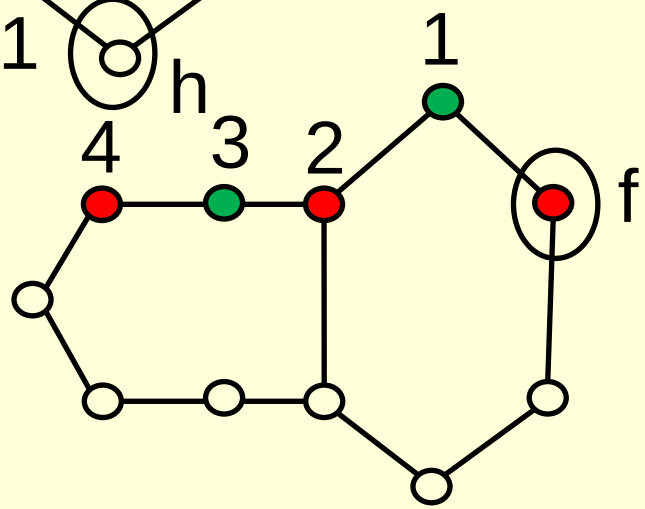
T f4b3e



T' f4b1k3h



K' f4b3e2g

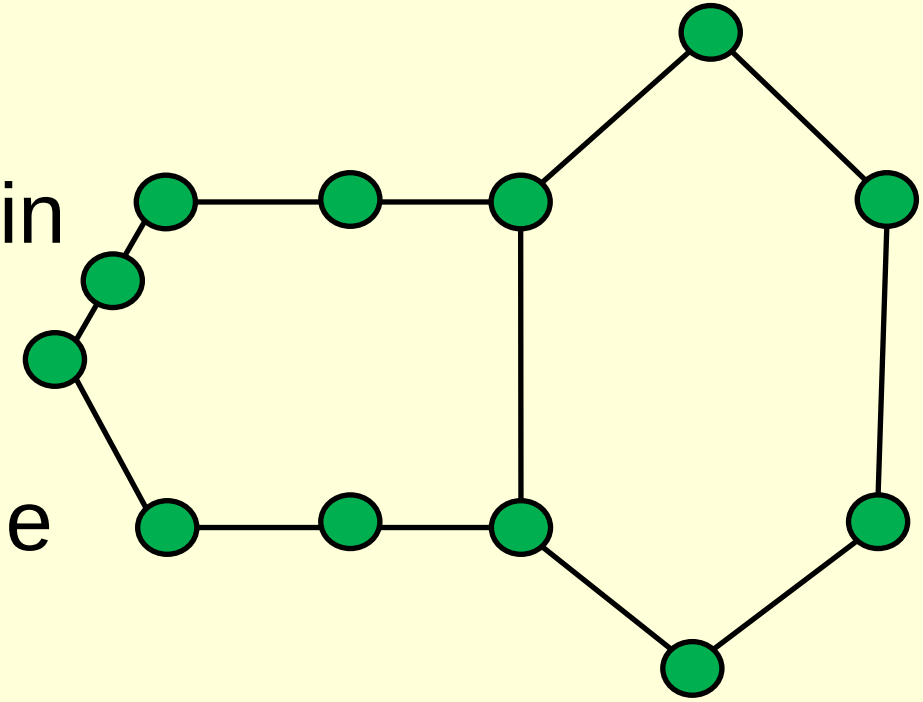


K f4b1k3h

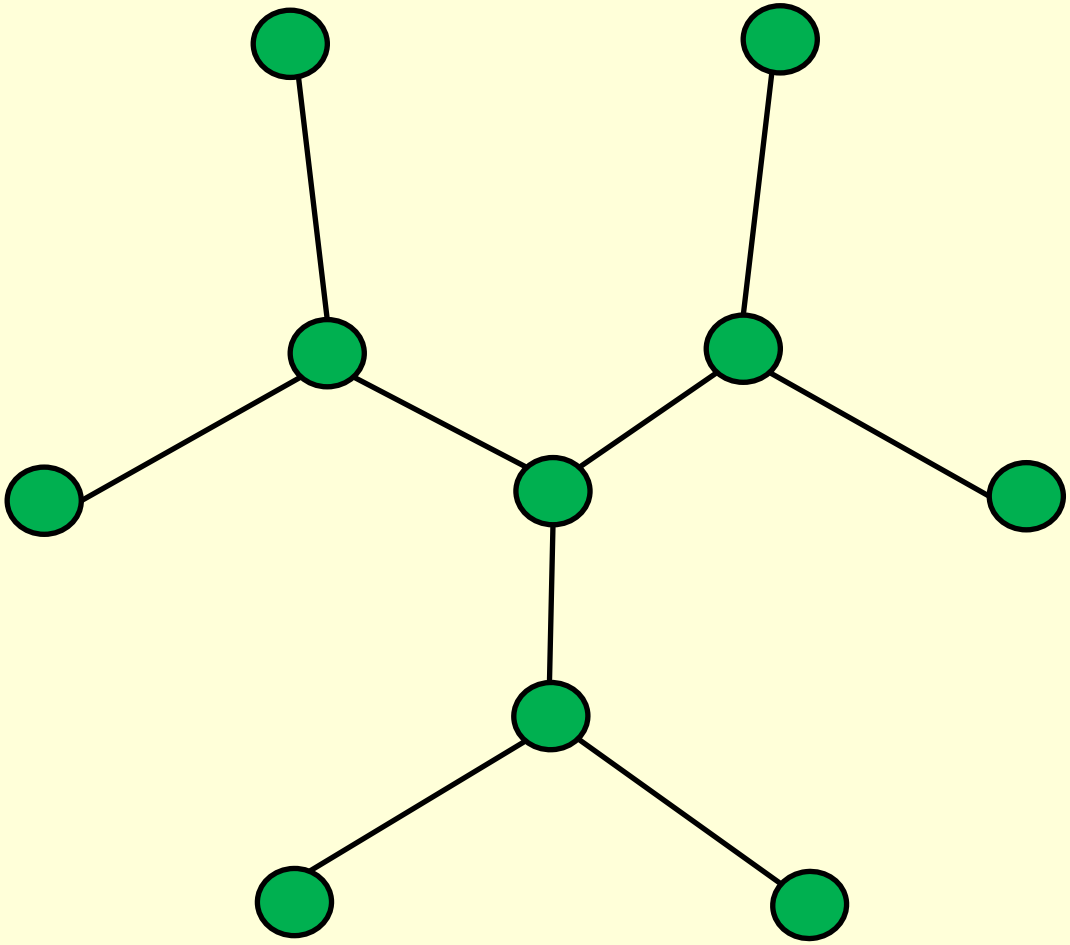
An odd cycle is necessary (but not sufficient) for Cop to win on a graph with a 6-cycle.

Theorem. A 6-cycle is a hideout on any graph such that no edge of the 6-cycle is contained in an odd cycle.

Note that this Robber-win graph is obtained from the previous Cop-win graph by subdividing one edge.



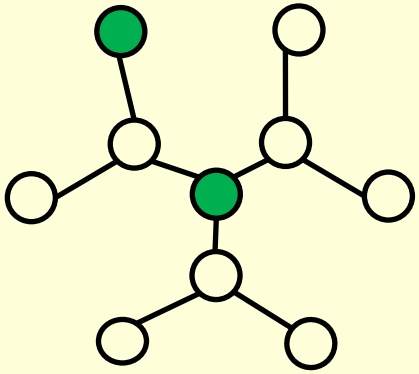
$T_{3,3}$: A hideout in any tree



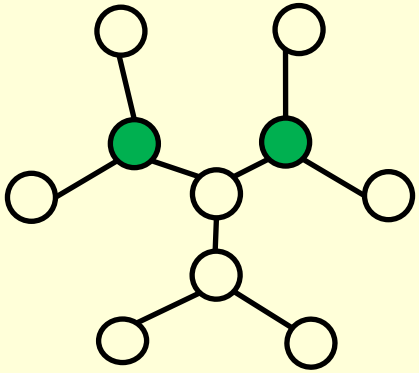
$T_{3,3}$: A hideout in any tree

Claim: after each probe, there is a distance at which Robber could be at the center and at least one leaf, or else at two or more neighbours of the centre.

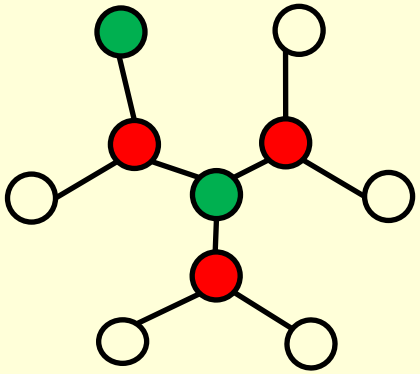
That is,



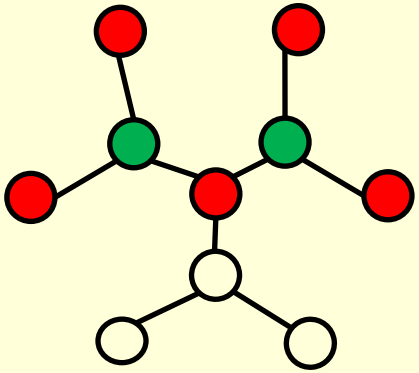
or



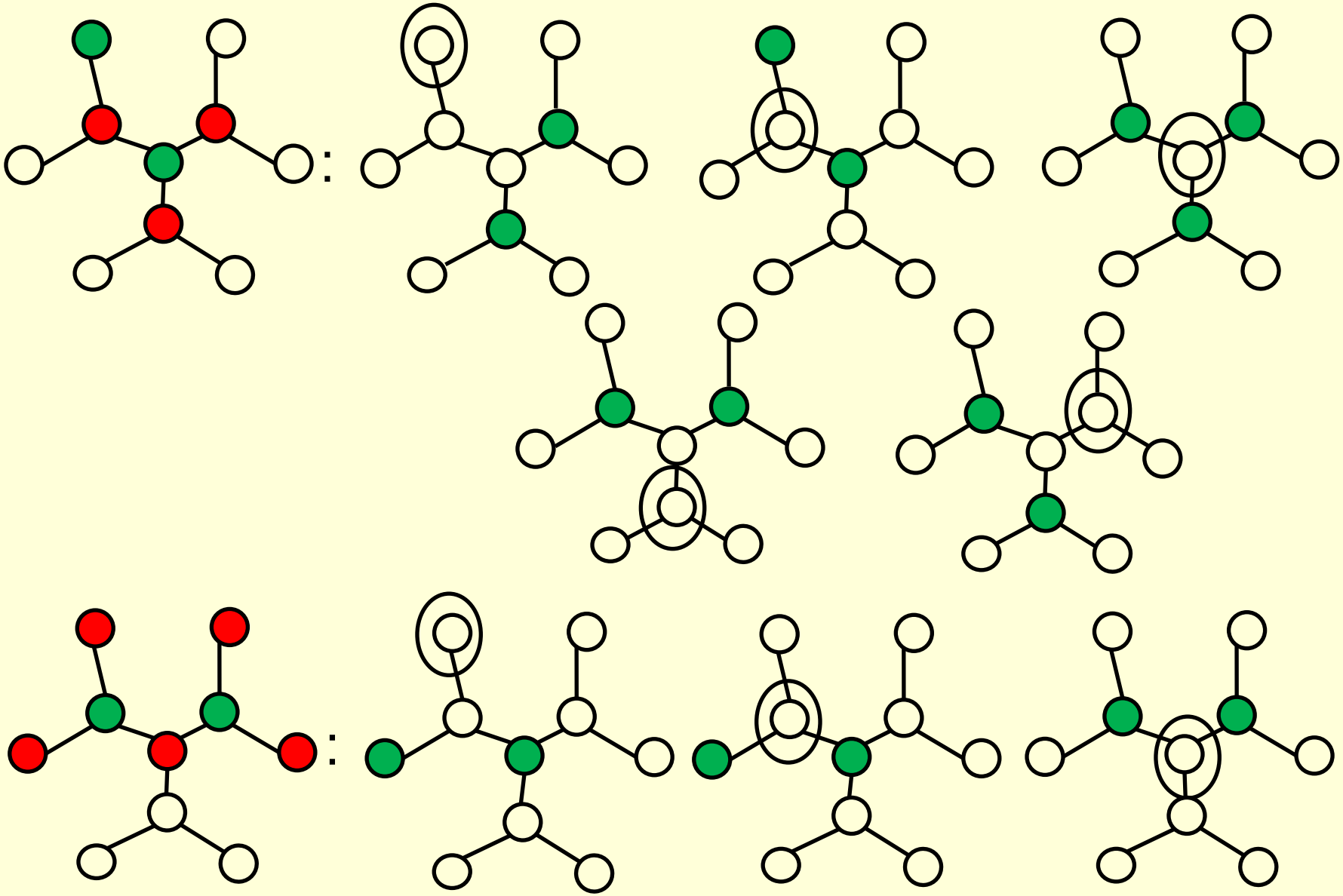
After move,



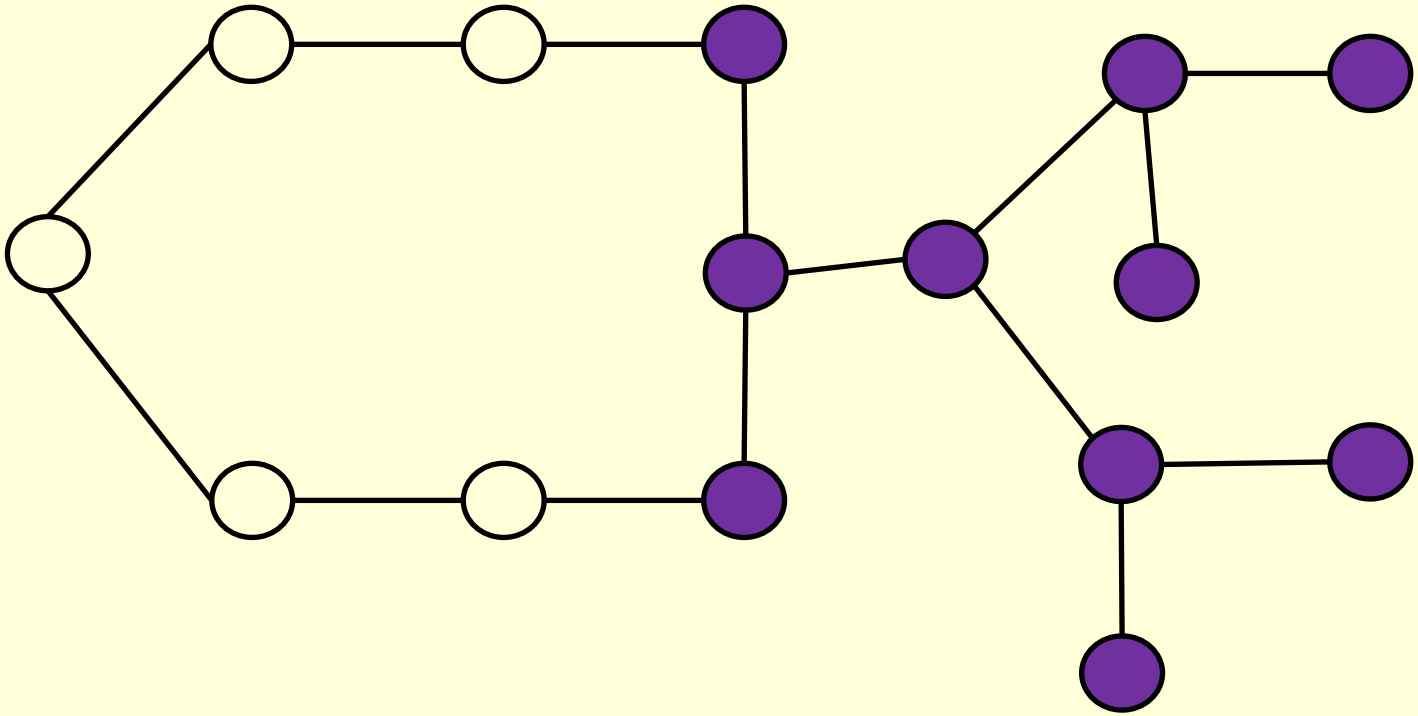
or

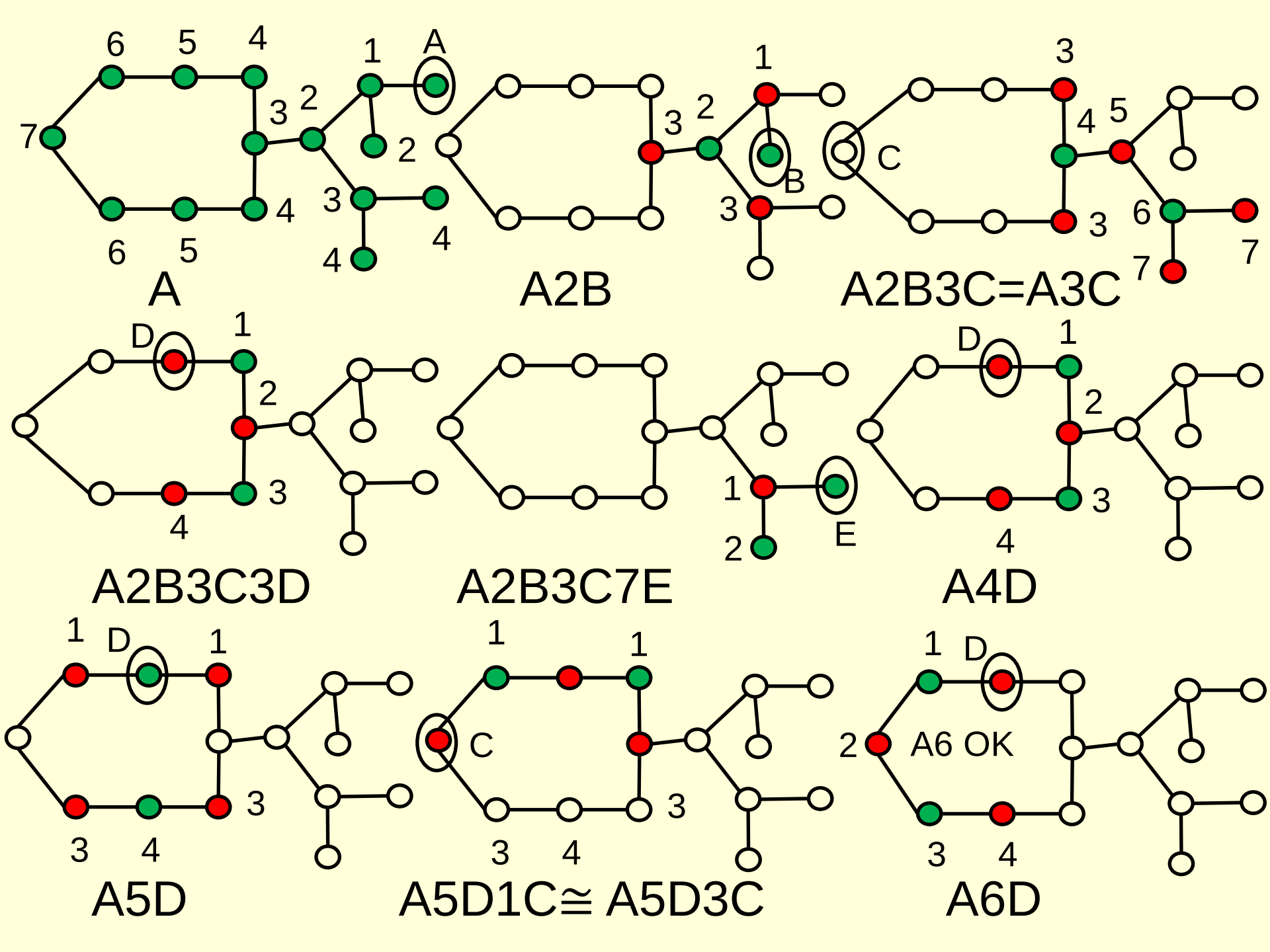


$T_{3,3}$: A hideout in any tree



$T_{3,3}$: A hideout in any tree but not necessarily in non-trees:

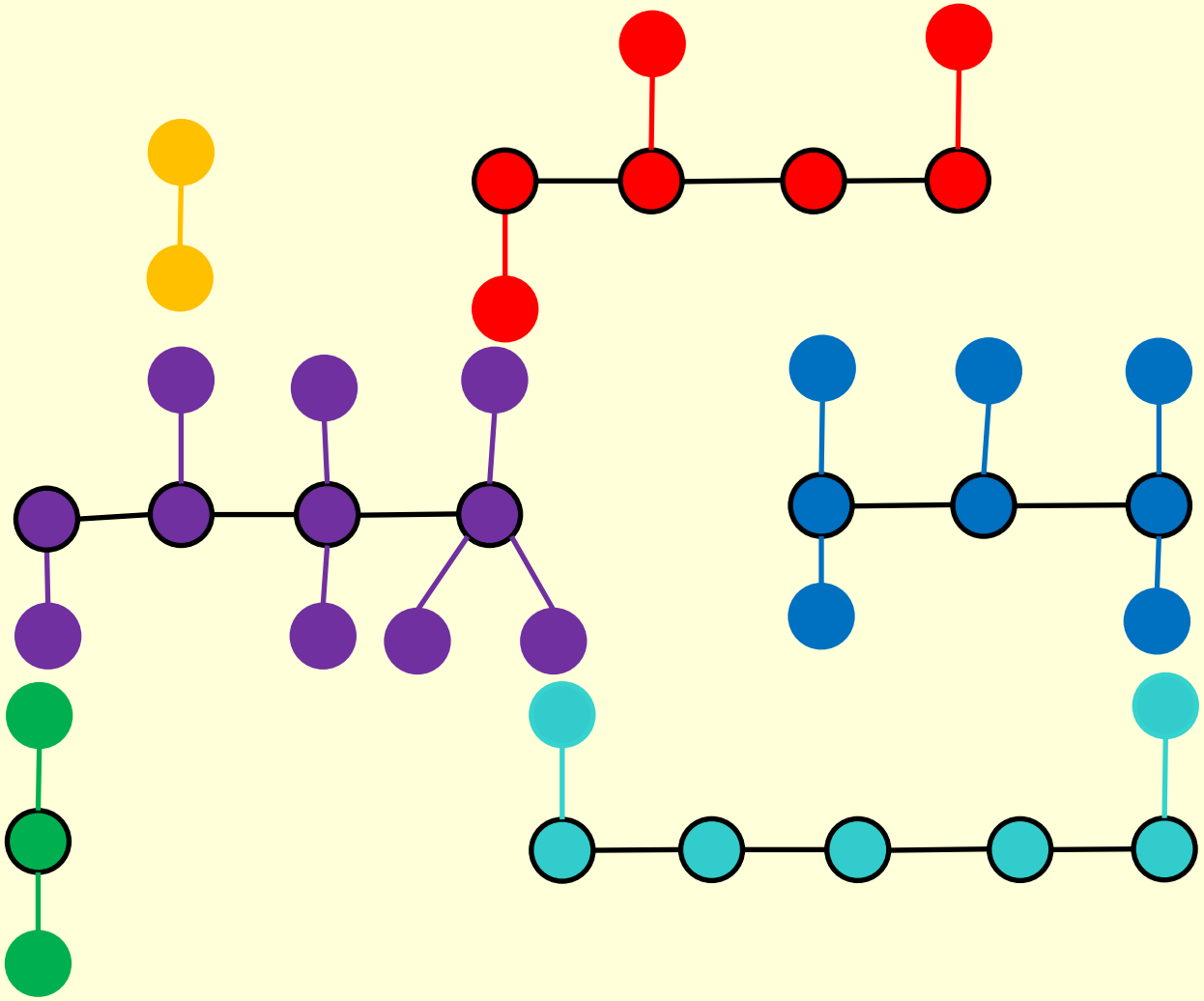



A
A2B
A2B3C=A3C
A2B3C3D
A2B3C7E
A4D
A5D
A5D1C≅ A5D3C
A6D
A6 OK

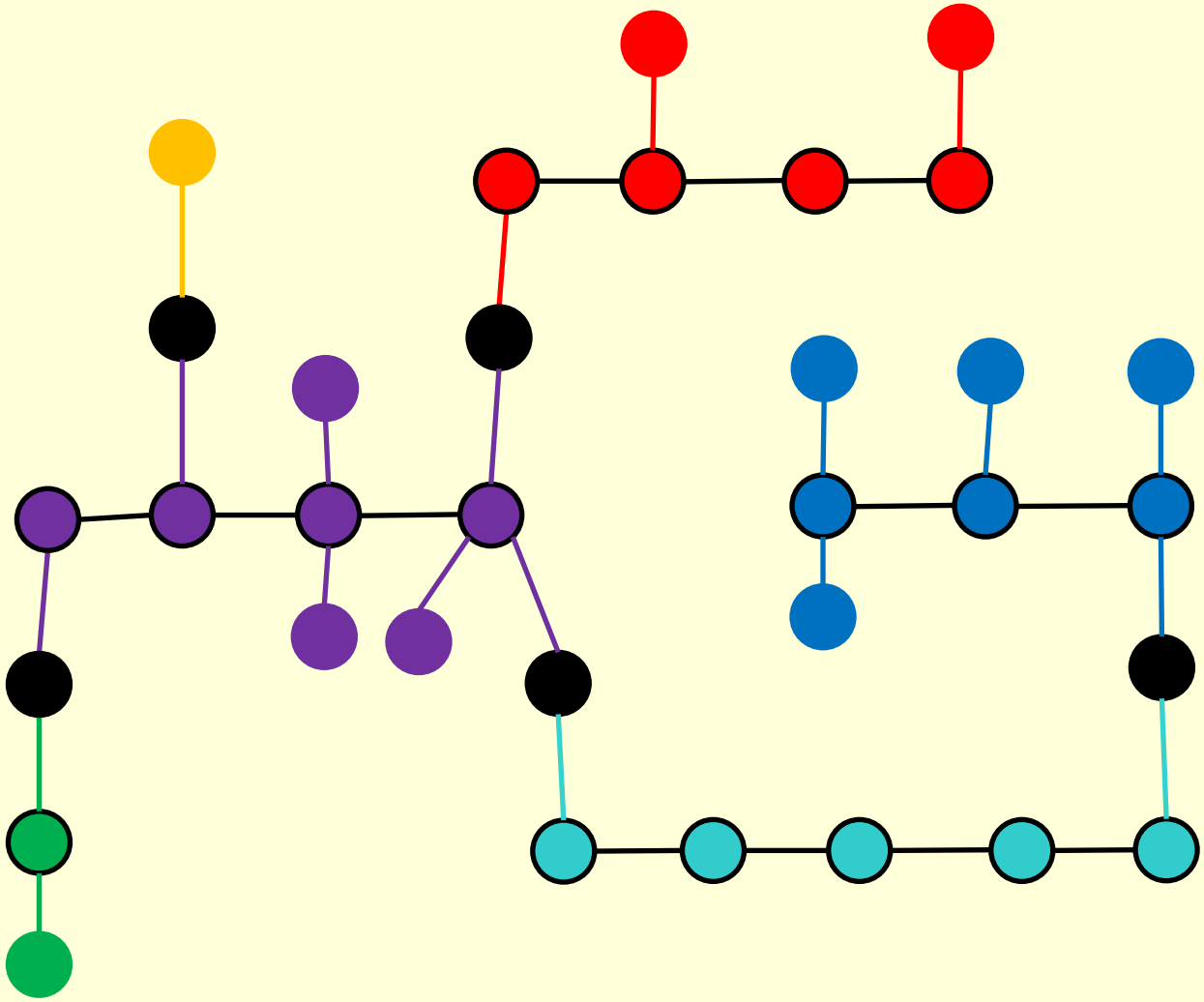
On which trees does Cop win?

- In fact Cop wins on **ALL** trees with no $T_{3,3}$ subtree.
- Proof is a strategy algorithm of the same flavour but much longer than those shown here.
- These trees can be characterized as trees of the form "**Caterpillars shaking feet**".

Caterpillars

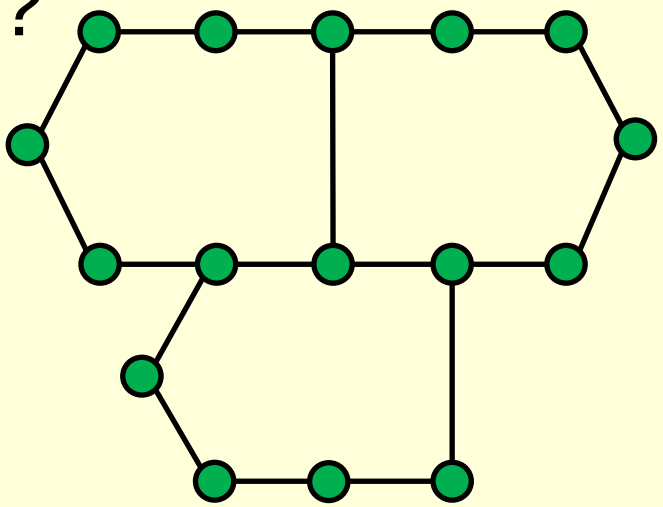


Caterpillars Shaking Feet



Where next?

- Carraher et al. [2012] studied bounds on the number of subdivisions of each edge required so that Cop wins (always exist)
- Both larger cycles and longer threads (paths of vertices of degree 2) tend to Cop winning; how do these interact?



Thank you!