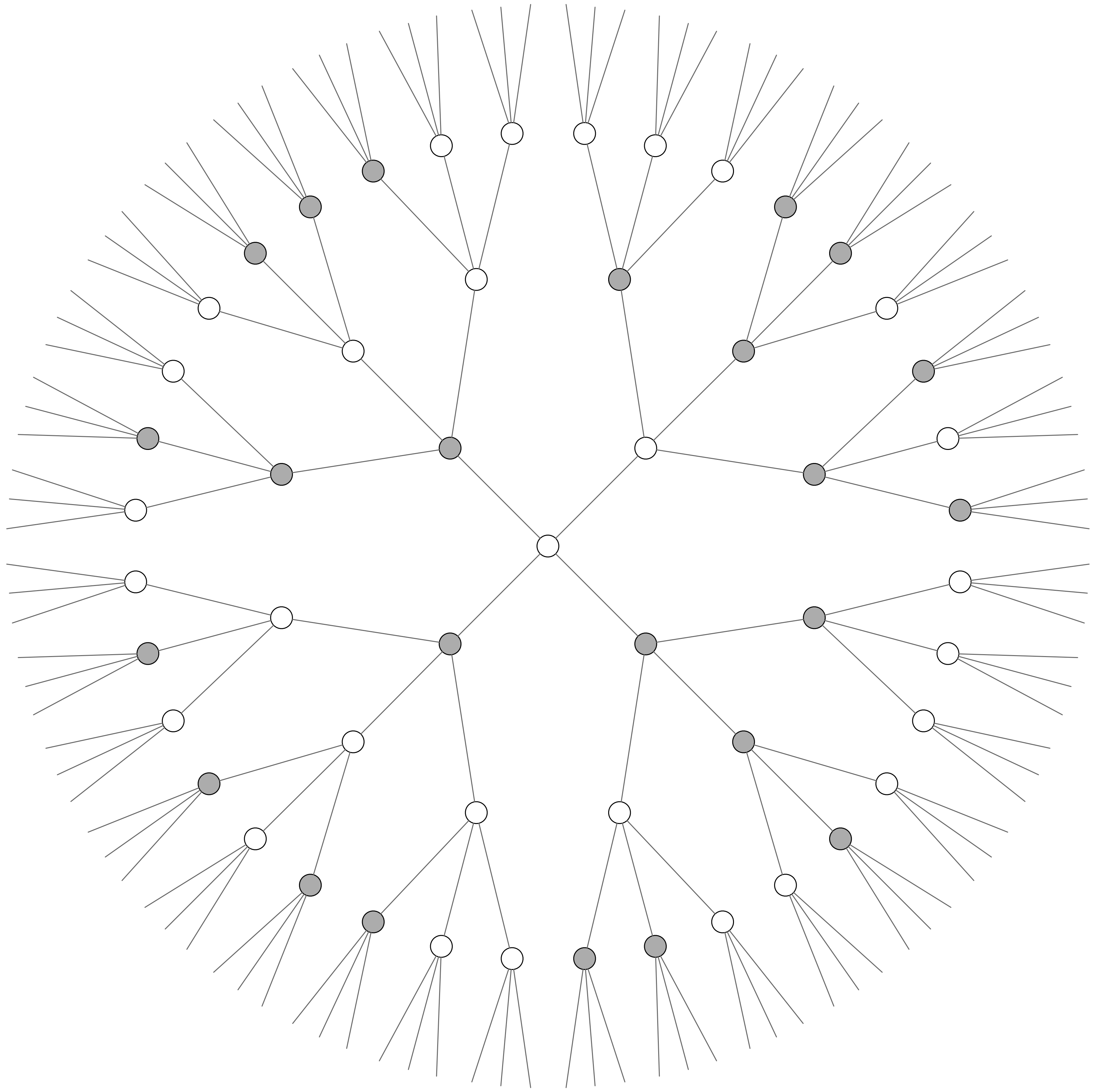
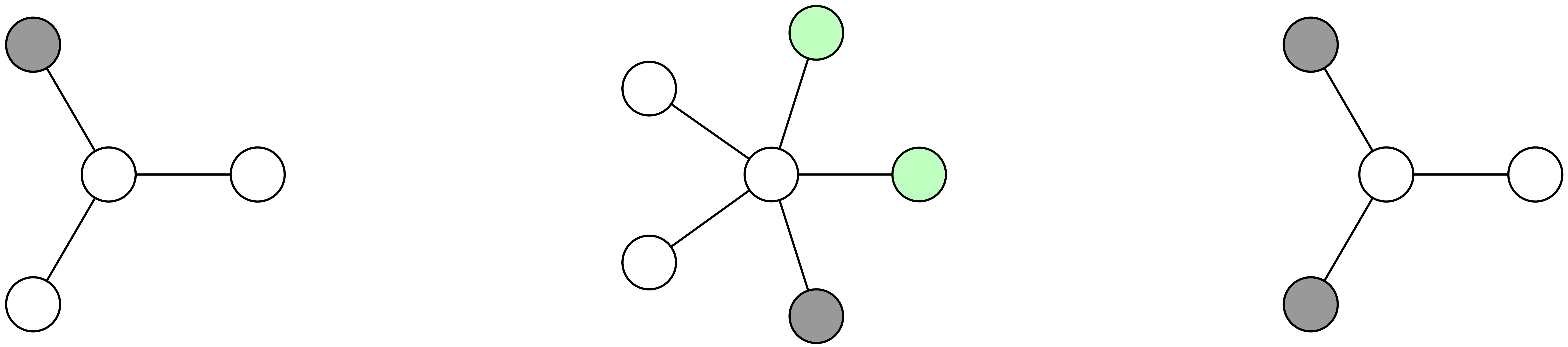
Locality of not-so-weak coloring

Alkida Balliu · Aalto University
Juho Hirvonen · Aalto University
Christoph Lenzen · MPI
Dennis Olivetti · Aalto University
Jukka Suomela · Aalto University



General topic

Upper bounds and **lower** bounds for relaxed **versions of vertex coloring**



LOCAL model

- Entities = **nodes**
- Communication links = **edges**
- Input graph = communication graph

LOCAL model

- Each node has a **unique identifier** from 1 to poly(n)
- **No bounds** on the computational power of the entities
- **No bounds** on the bandwidth

LOCAL model

- Round 0

LOCAL model

- Round 1

LOCAL model

- Round 2

LOCAL model

- After *t* **rounds**: knowledge of the graph up to **distance *t***
- Focus on **locality**

Locally Checkable Labelings (LCLs)

- **Input**
  - Graph of **constant** maximum degree Δ
  - Node labels from a **constant-size** set X
- **Output**
  - Node labels from a **constant-size** set Y, such that each node satisfies some **local constraints**
- **Correctness**
  - A solution is globally correct if it is correct in all **constant-radius** neighborhoods

[Naor and Stockmeyer, 1995]

Example: vertex k-coloring

- **Output**: color nodes from a palette of ***k* = *O*(1) colors**
- **Constraint**: each node must have a **different color** from its neighbors

Example: weak 2-coloring

- **Output**: color nodes from a palette of **2 colors**
- **Constraint**: each node must have a **different color** from **at least 1** neighbor

“Easy” and “hard” LCLs

Fix an LCL: its deterministic distributed complexity is either $O(\log^* n)$ or at least $\Omega(\log n)$ [Chang et al., 2016]

- “**Easy**”: LCLs solvable in $O(\log^* n)$ rounds
- “**Hard**”: LCLs that require at least $\Omega(\log n)$ rounds

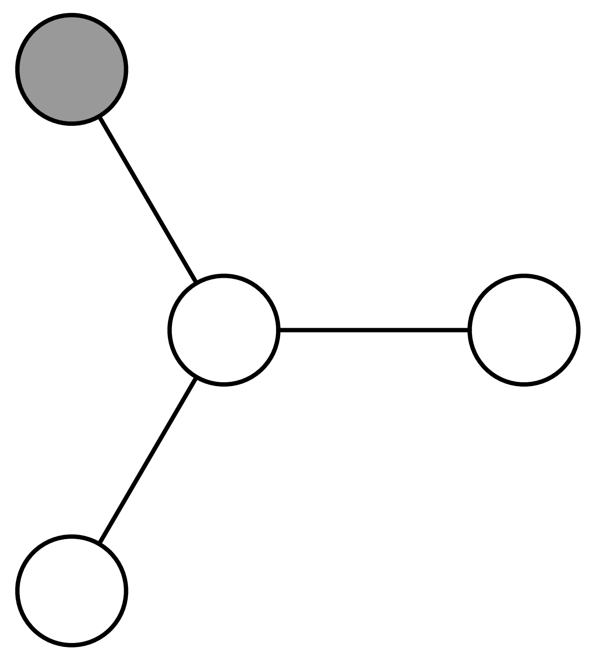
From easy to hard: edge coloring

- $(2\Delta - 1)$-edge-coloring is **easy**
- $(2\Delta - 2)$-edge-coloring is **hard**

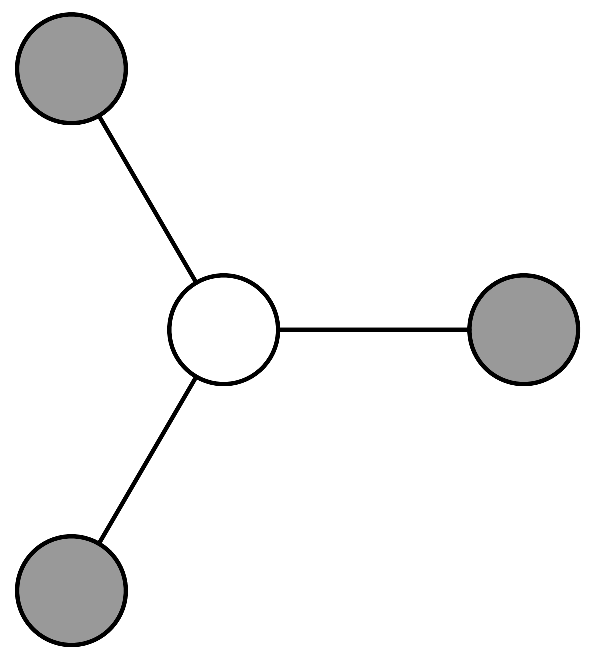
From easy to hard: vertex coloring

- $(\Delta + 1)$-vertex-coloring is **easy**
- Δ-vertex-coloring is **hard**

From easy to hard

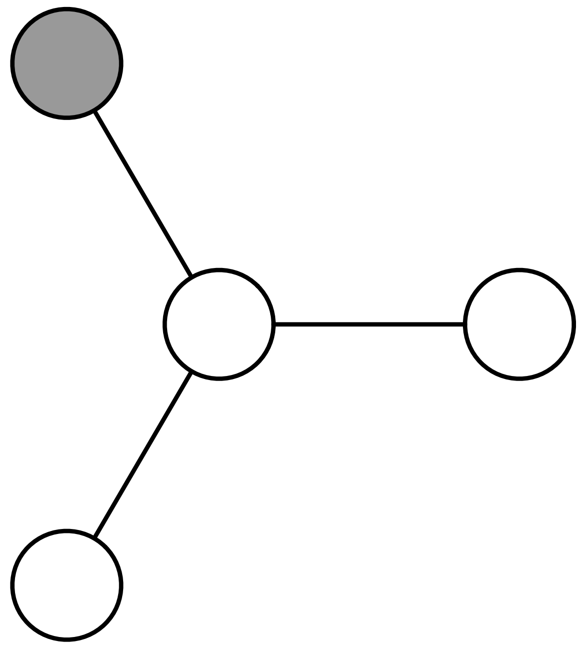


Weak 2-coloring Easy

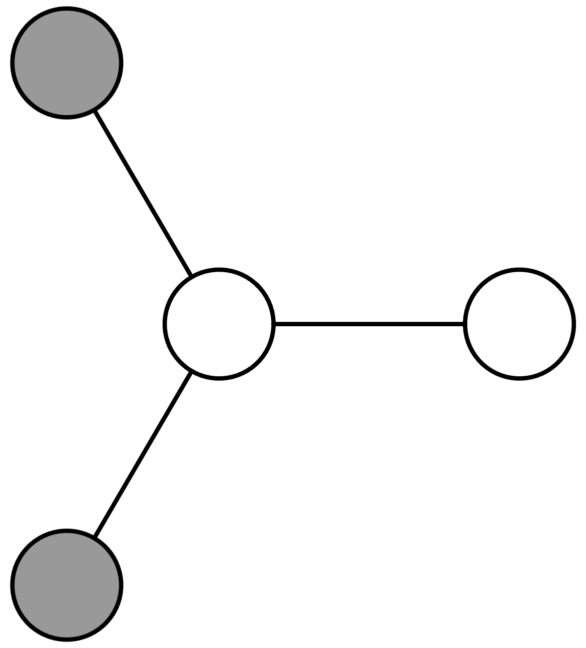


2-coloring Hard

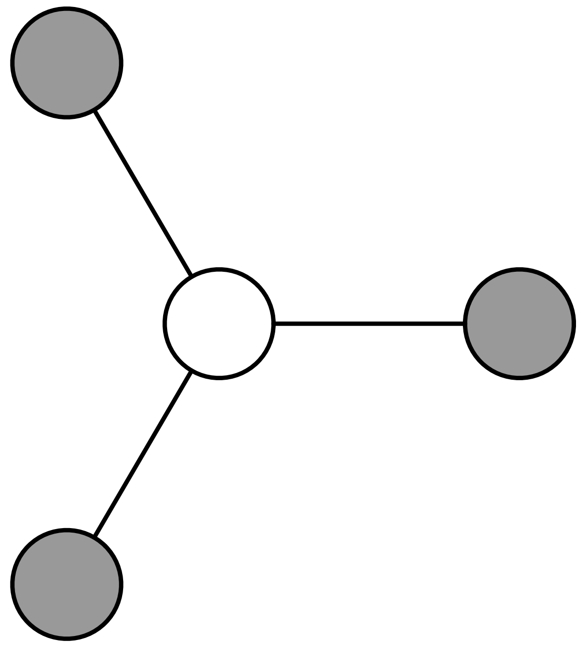
From easy to hard



Weak 2-coloring Easy



Intermediate ?



2-coloring Hard

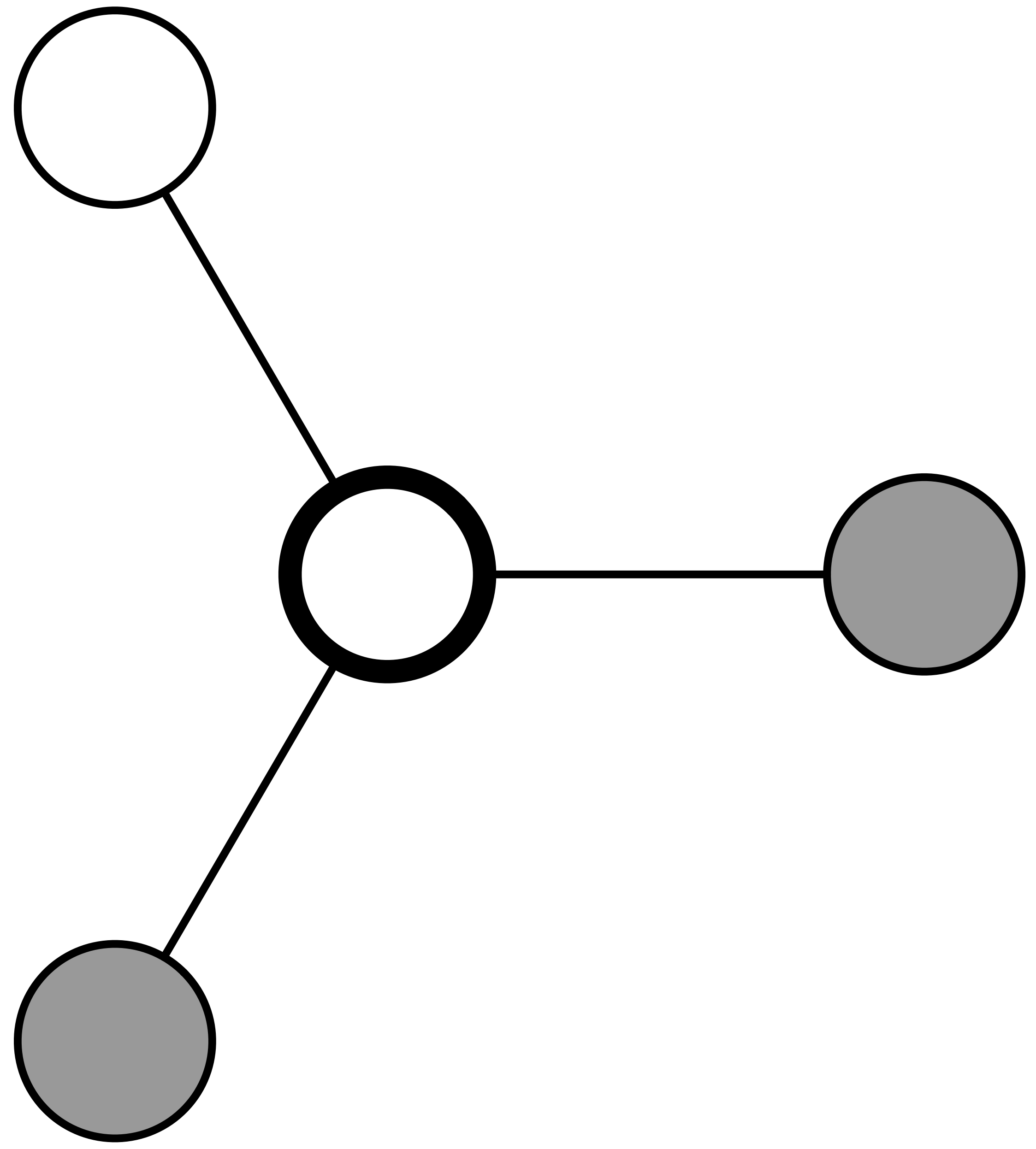
Partial coloring

- **Input**: graph of **minimum degree *d***
- **Output**: label nodes from a palette of **c colors**
- **Constraint**: each node v must have **at least *k*** neighbors having a **different** color from v

For which values of (k, c, d) is this problem easy? For which ones is it hard? We study this problem in trees and general graphs.

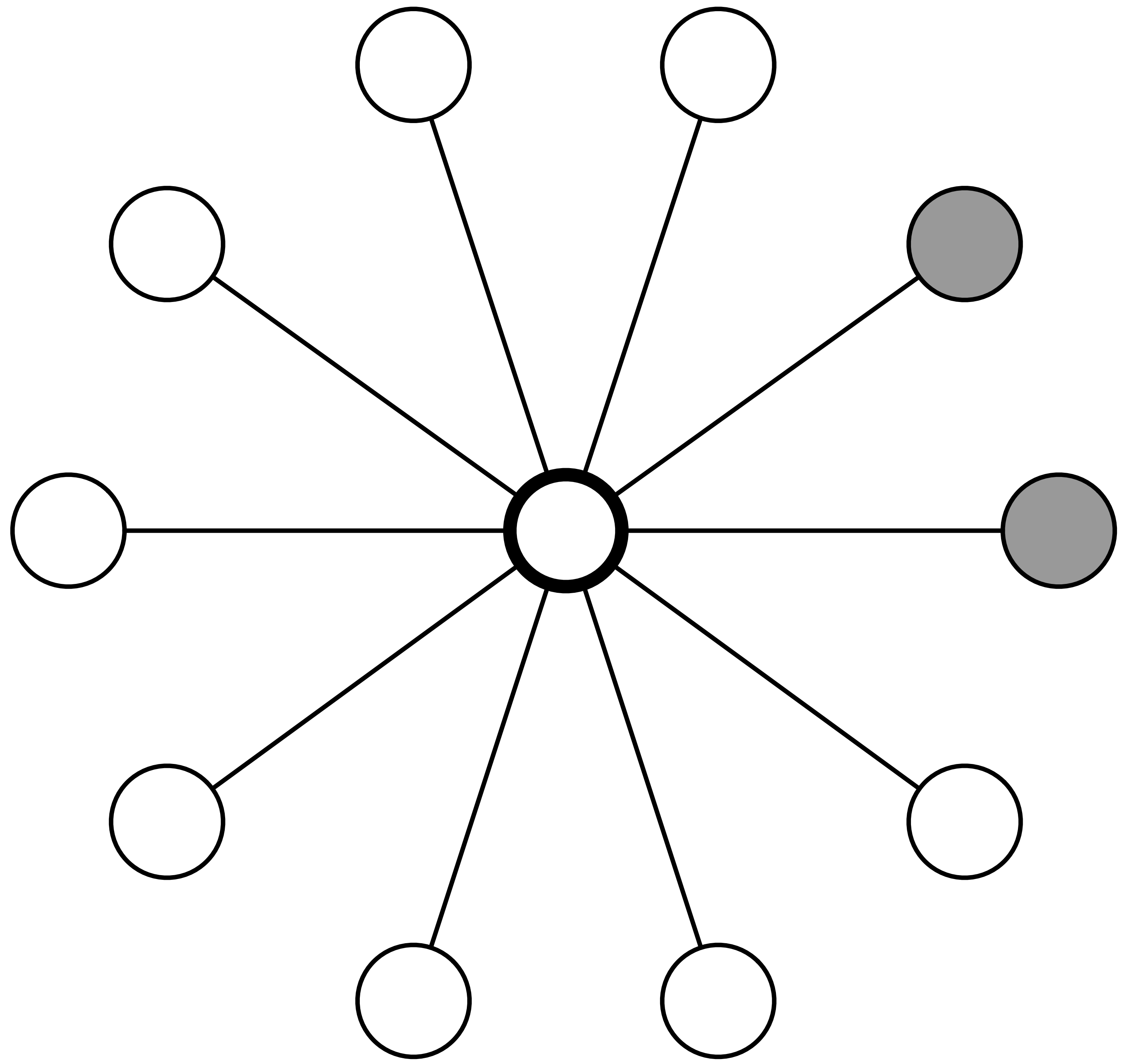
Examples of partial colorings

- 2-partial 2-coloring
  - Palette of 2 colors
  - **3-regular** tree
  - Each node v must have **at least 2** neighbors having color different from v



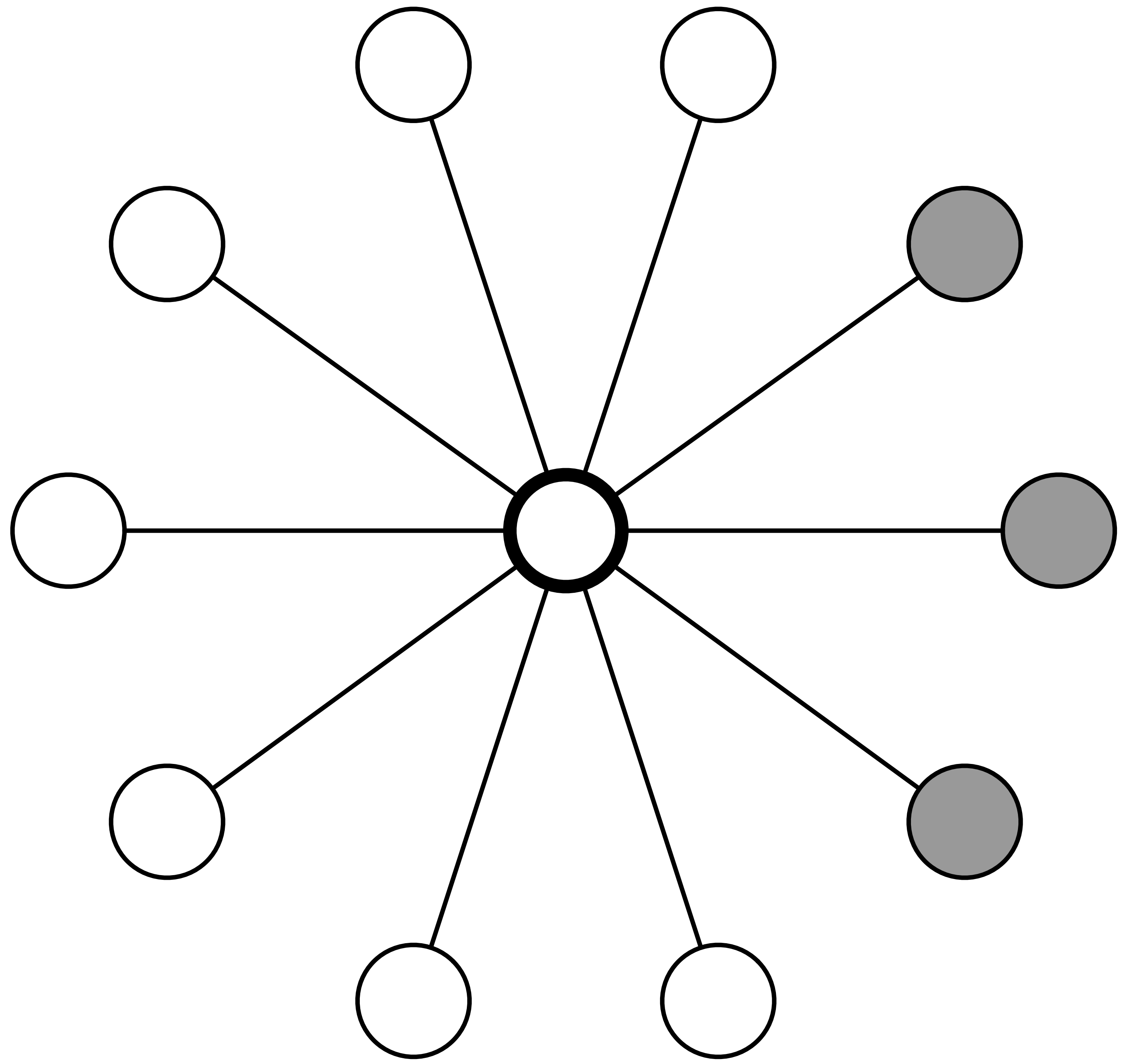
Examples of partial colorings

- 2-partial 2-coloring
  - Palette of 2 colors
  - **10-regular** tree
  - Each node v must have **at least 2** neighbors having color different from v



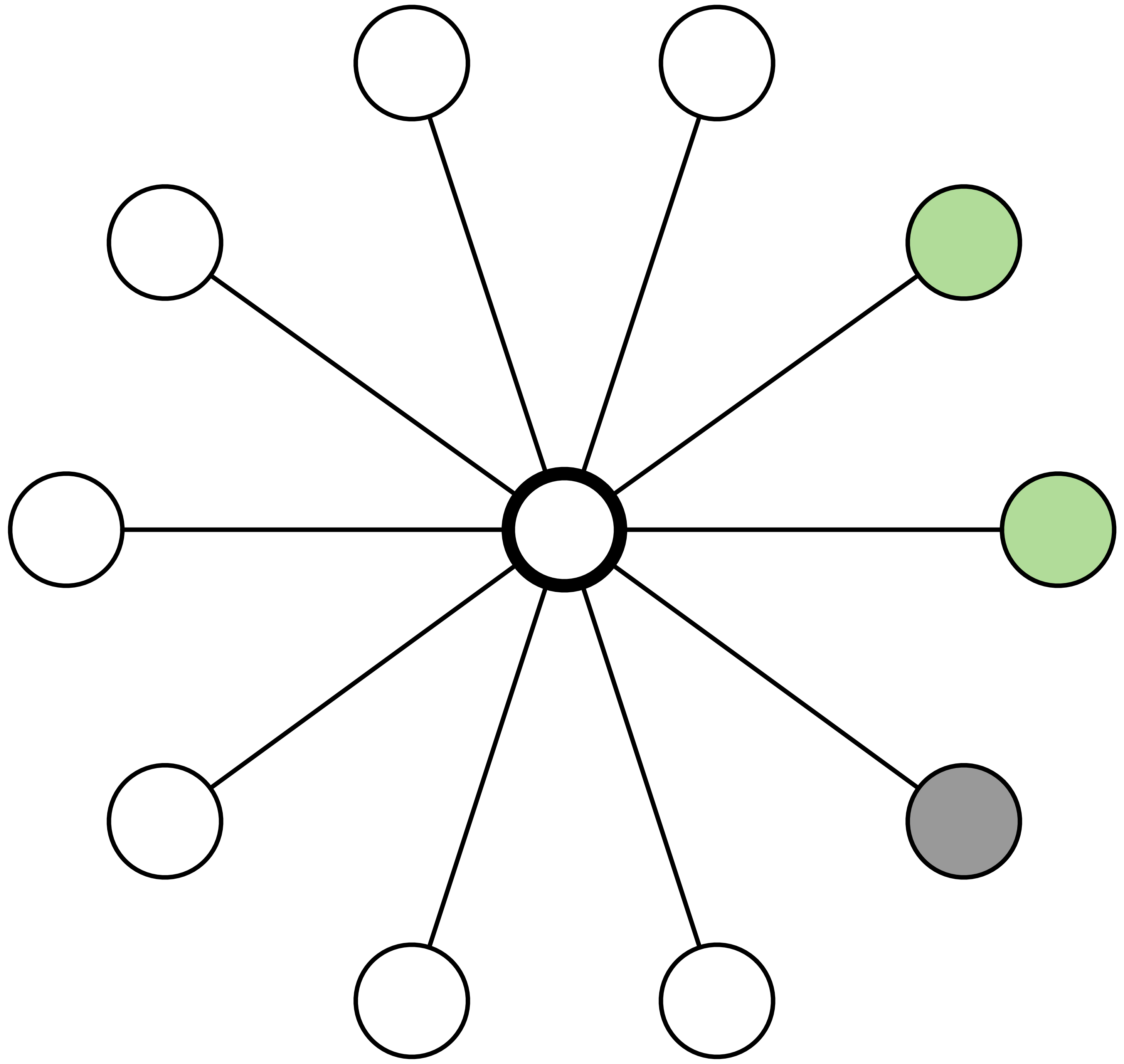
Examples of partial colorings

- 3-partial 2-coloring
  - Palette of 2 colors
  - **10-regular** tree
  - Each node v must have **at least 3** neighbors having color different from v



Examples of partial colorings

- 3-partial 3-coloring
  - Palette of 3 colors
  - **10-regular** tree
  - Each node v must have at least 3 neighbors having color different from v



Partial vs defective coloring

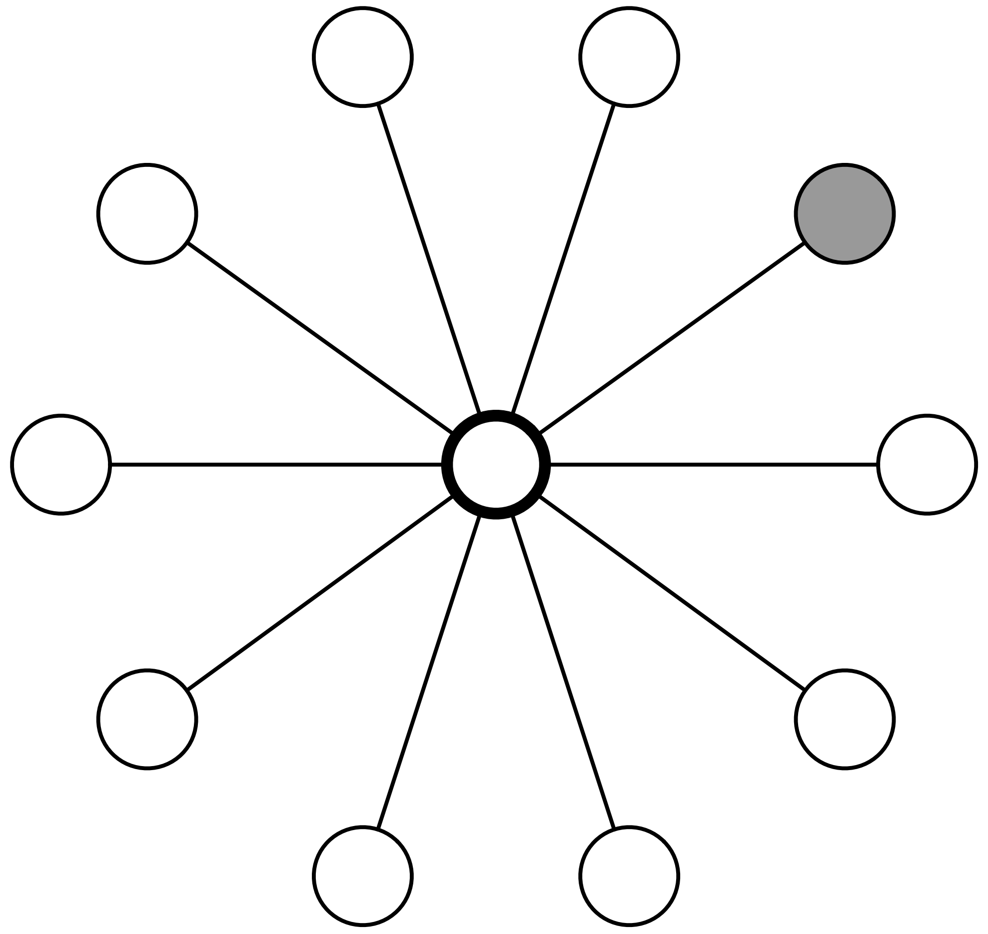
- **Partial coloring**:
    - each node v must have **at least k** neighbors having a **different** color from v
- **Defective coloring**:
    - each node v must have **at most k'** neighbors having the **same** color as v
- In d-regular graphs:
    - **k-partial c-coloring** = **$(d - k)$-defective c-coloring**

easy for some "small" k when c = perfect square

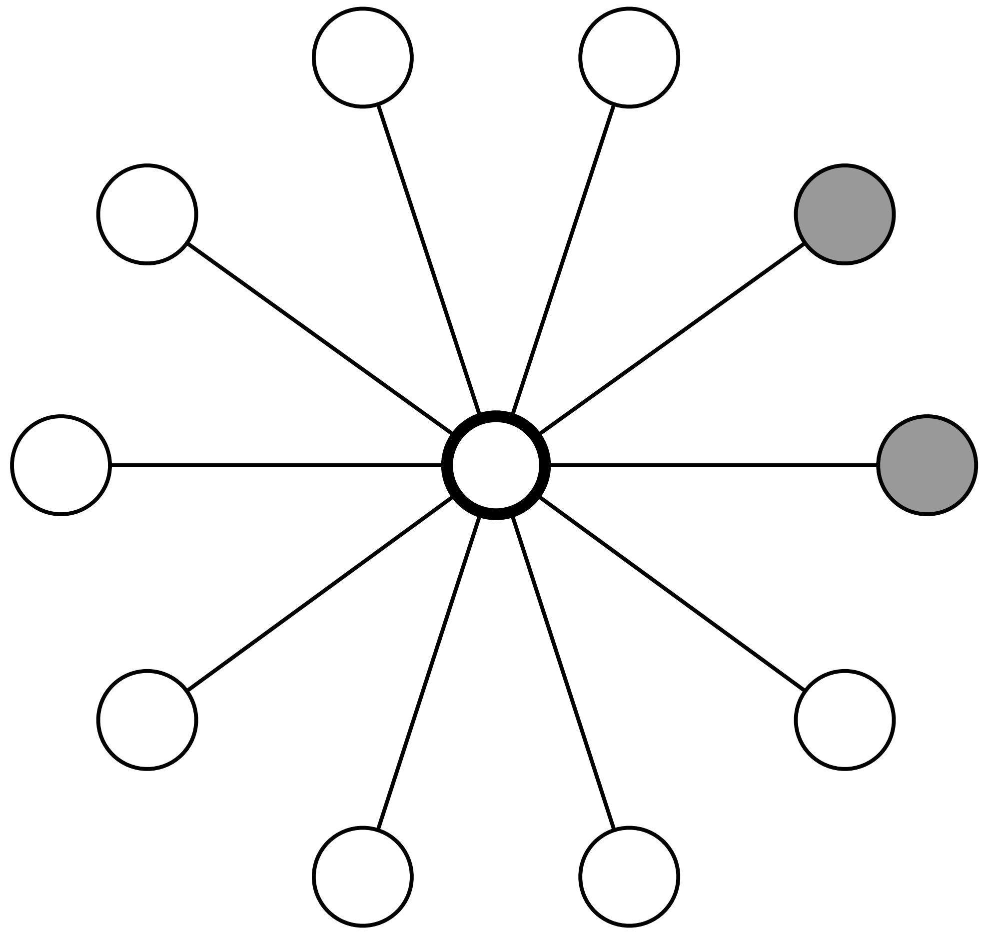
[Barenboim et al., 2014]

Weak 2-coloring and beyond

1-partial 2-coloring (weak 2-coloring) **Easy**



2-partial 2-coloring **?**



Our results: 2-partial 2-coloring is hard

2-partial 2-coloring in ***d***-regular trees requires at least $\Omega(\log n)$ rounds, $\forall\, d \geq 3$

- **Matching** $O(\log n)$ upper bound in trees [Bonamy et al., 2018]
- **Larger *d*** does not help!

Our results: 2-partial 2-coloring is hard

- Proof idea:
  - $o(\log n)$-round algorithm for 2-partial 2-coloring in d-regular trees
  - $O(1)$-round algorithm for sinkless orientation in $d^{O(1)}$-regular constant-distance-colored trees
  - $O(\log^* n)$-round algorithm for sinkless orientation in $d^{O(1)}$-regular trees
  - Contradiction [Brandt et al., 2016]

Our results: k-partial 3-coloring

- ***k***-partial **3**-coloring: the degree ***d*** plays an important role
  - hard for some “small” ***d***
  - easy for some “large” ***d***

Our results: k-partial c-coloring

k-partial 3-coloring in graphs with minimum degree $d = (3k - 4)$ is easy

k-partial *k*-coloring in graphs with minimum degree $d = (k + 2)$ is easy

k-partial *c*-coloring in *d*-regular graphs, for $k \geq d(c - 1)/c + 1$, is hard

Our algorithms are inspired by the ones for defective coloring of [Barenboim et al. 2014]

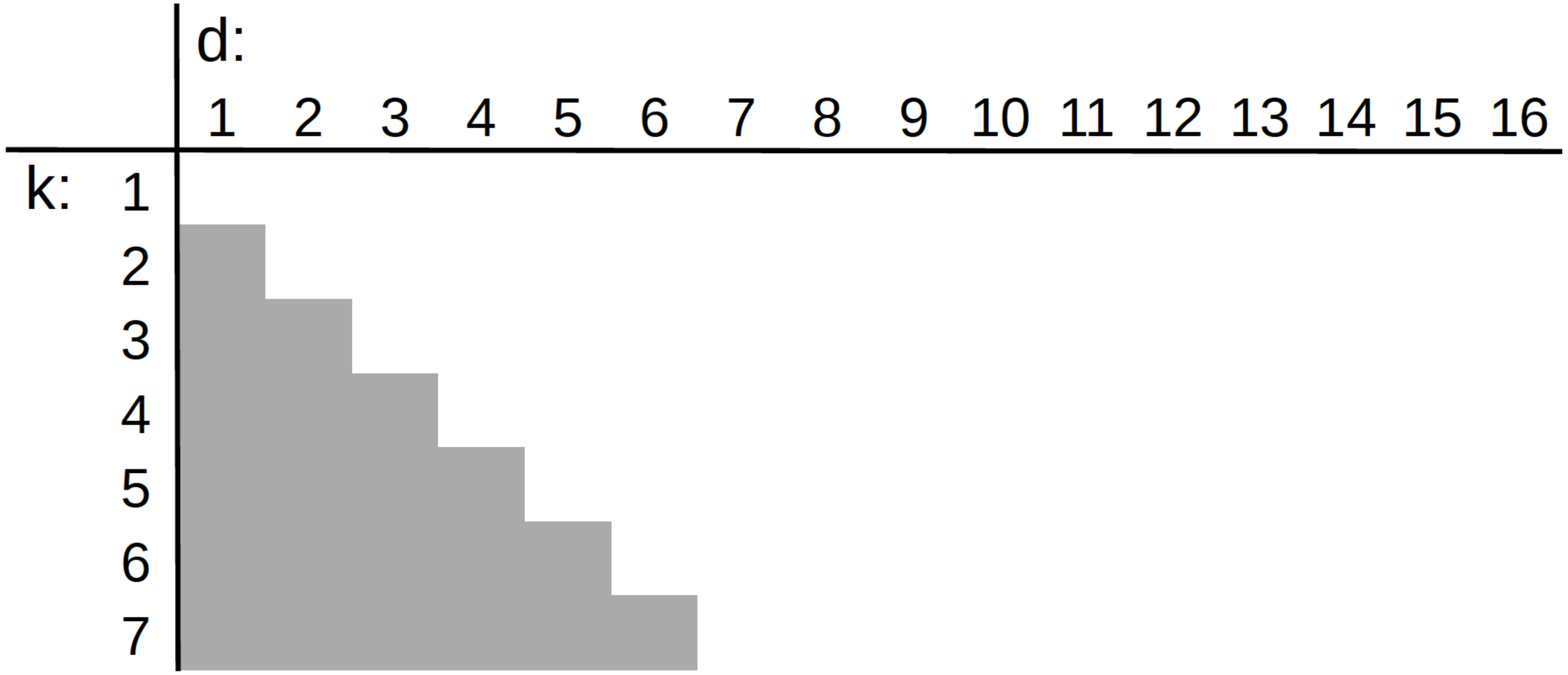
Summary

k*-partial 2-coloring in graphs with minimum degree *d

Easy

Hard

No solution



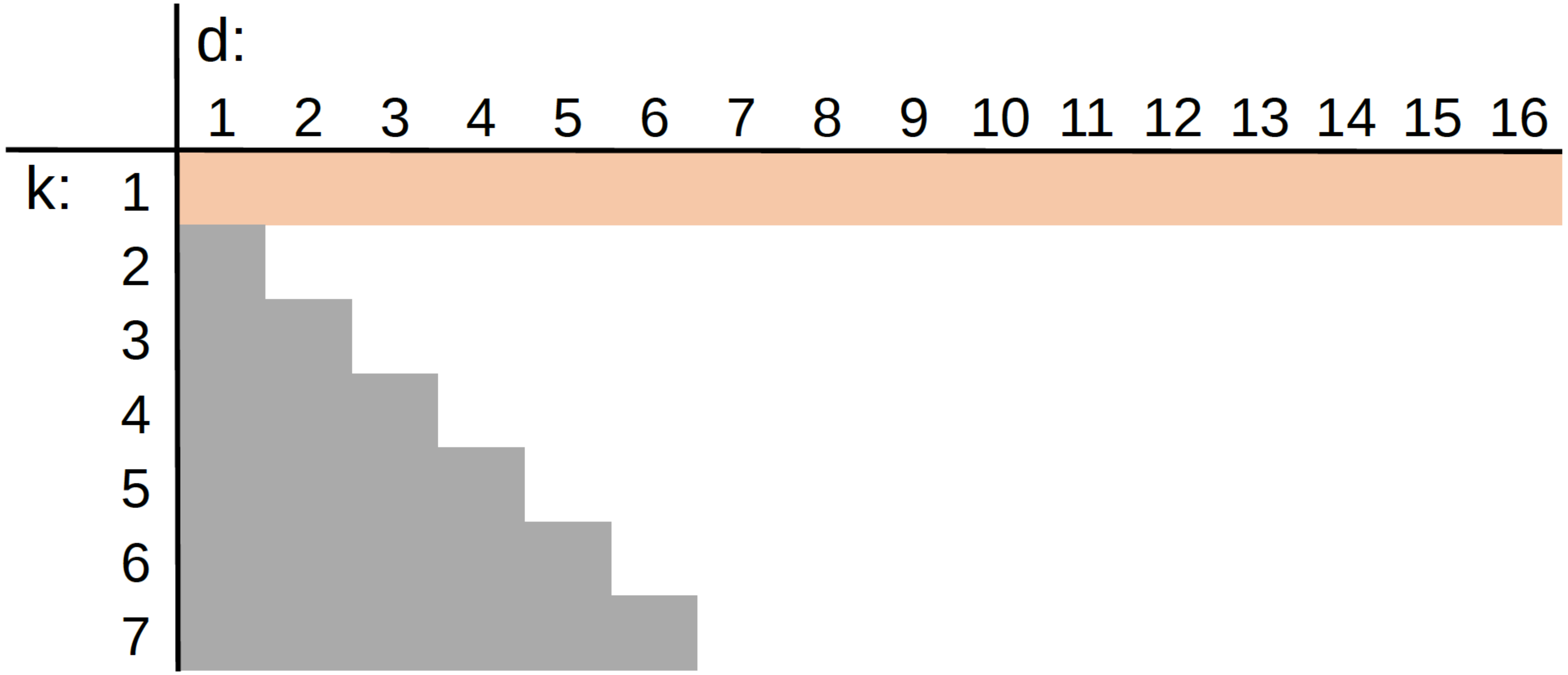
Summary

k*-partial 2-coloring in graphs with minimum degree *d

Easy

Hard

No solution



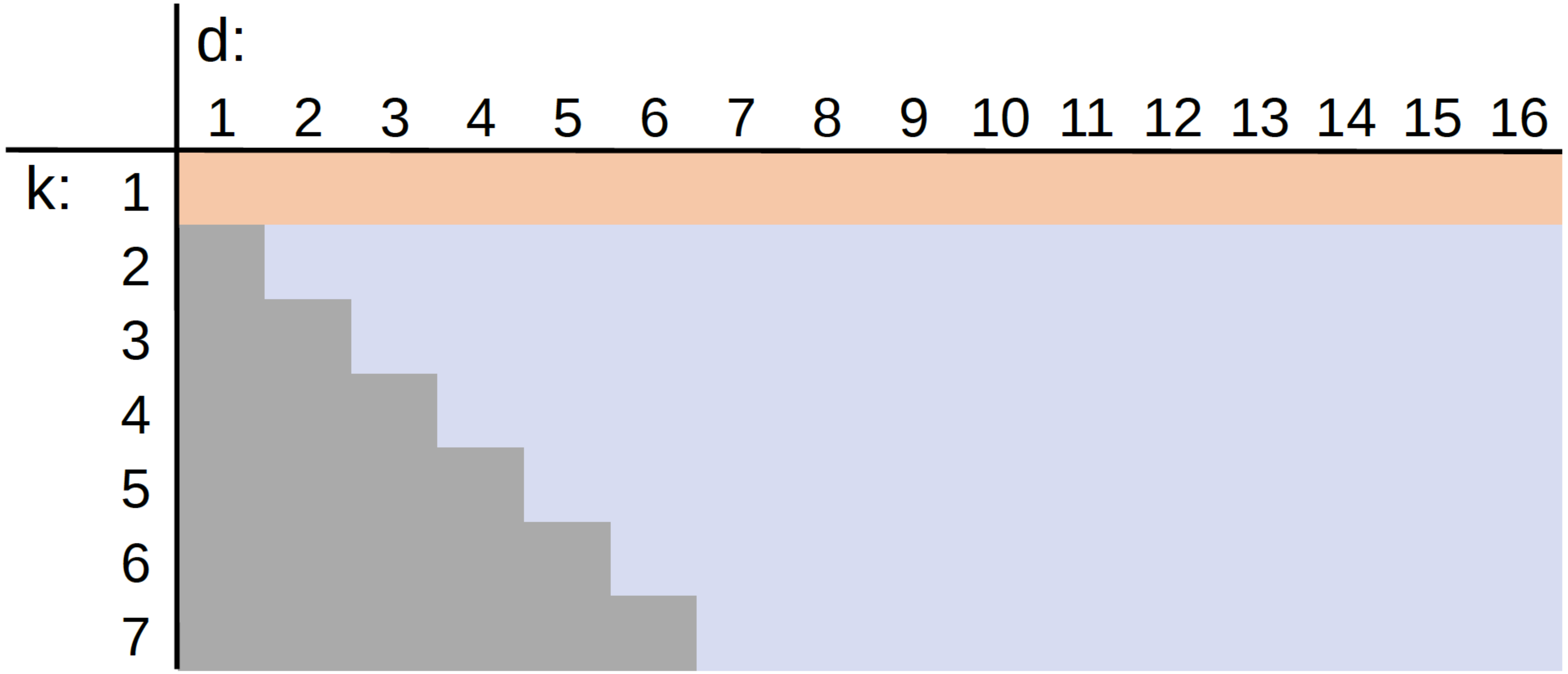
Summary

k*-partial 2-coloring in graphs with minimum degree *d

Easy

Hard

No solution



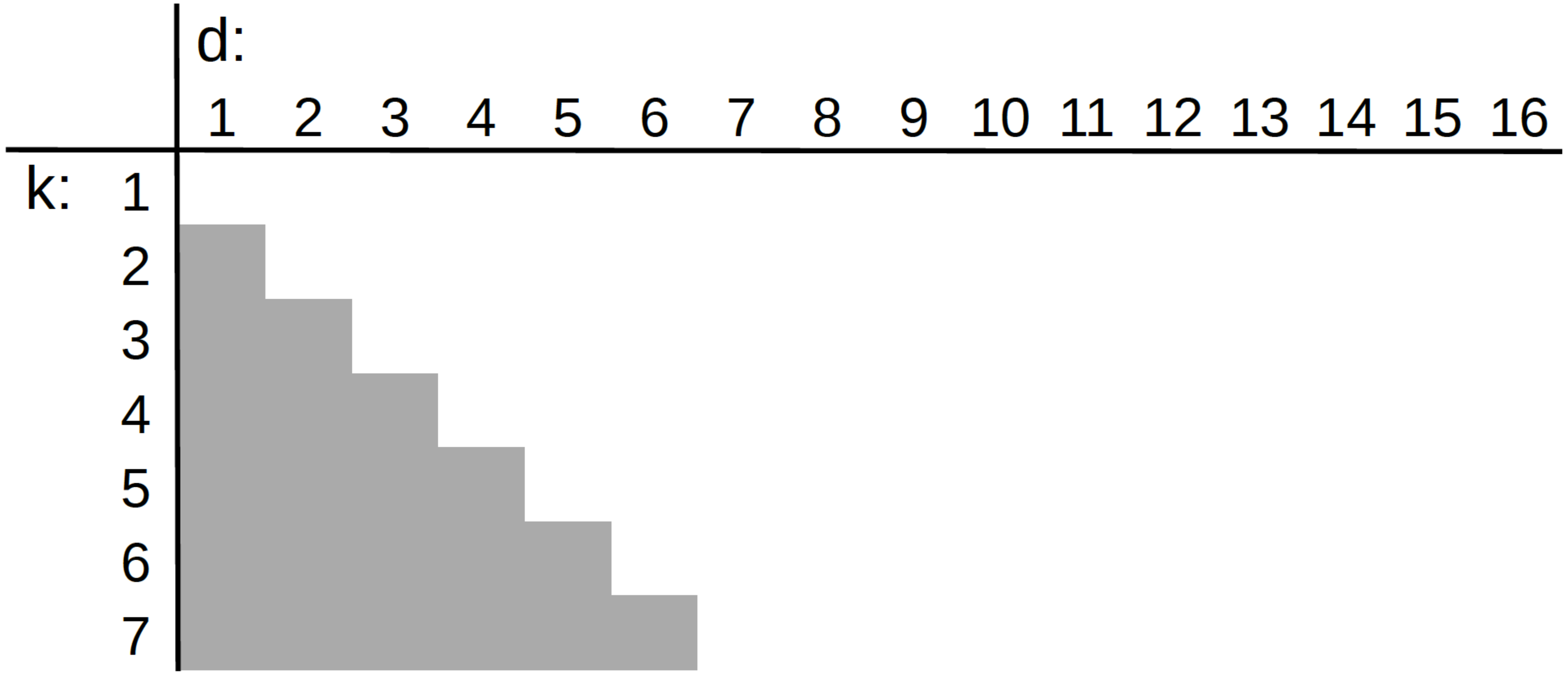
Summary

k*-partial 3-coloring in graphs with minimum degree *d

Easy

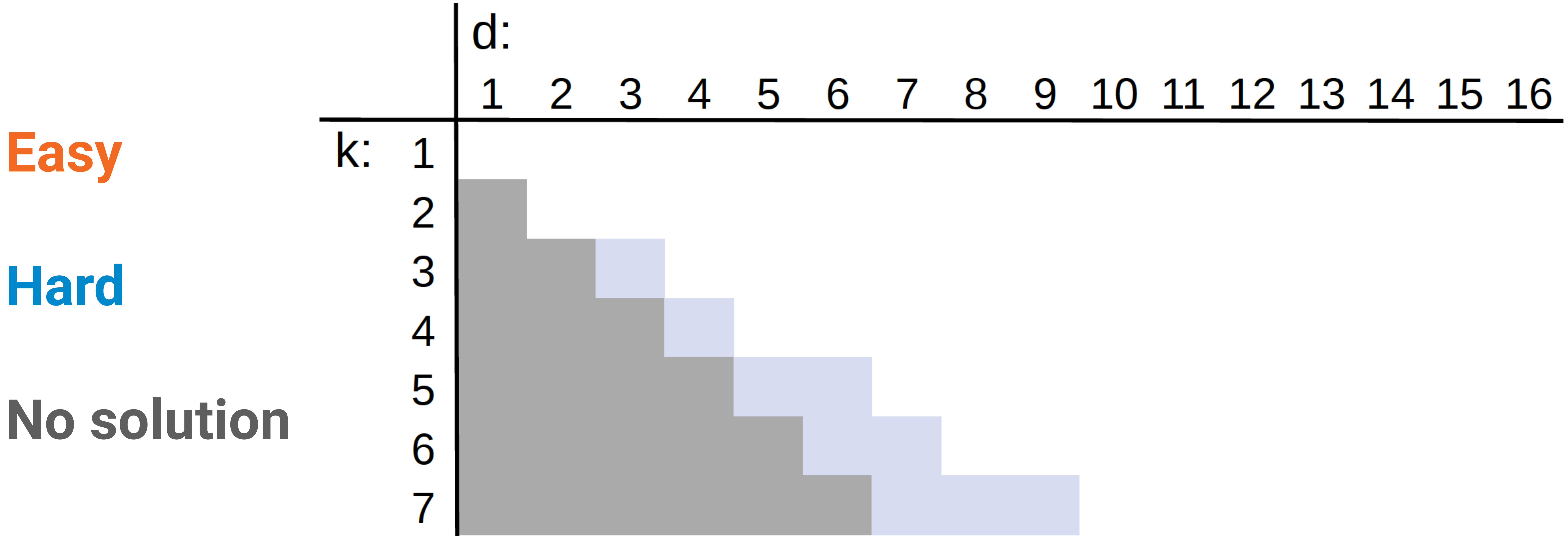
Hard

No solution



Summary

k-partial **3**-coloring in graphs with minimum degree ***d***



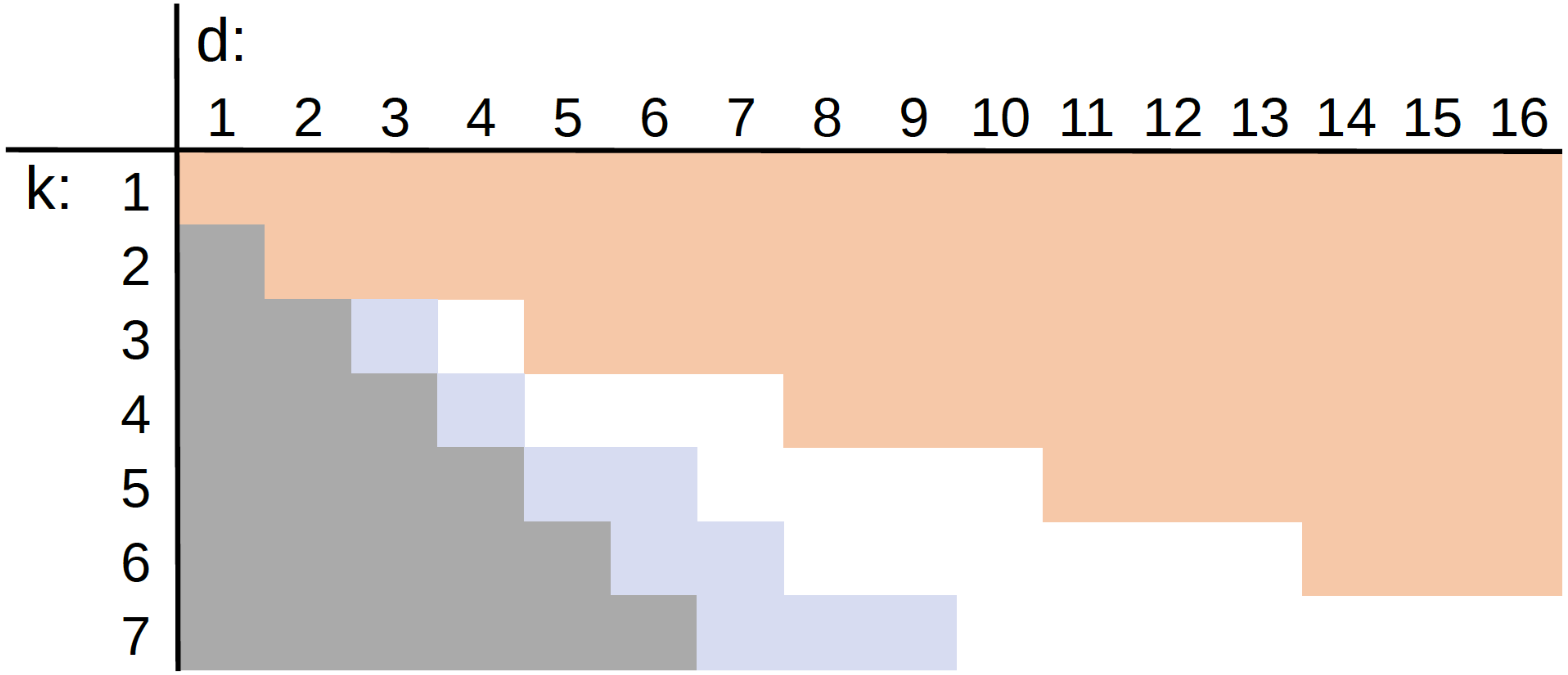
Summary

k*-partial 3-coloring in graphs with minimum degree *d

Easy

Hard

No solution



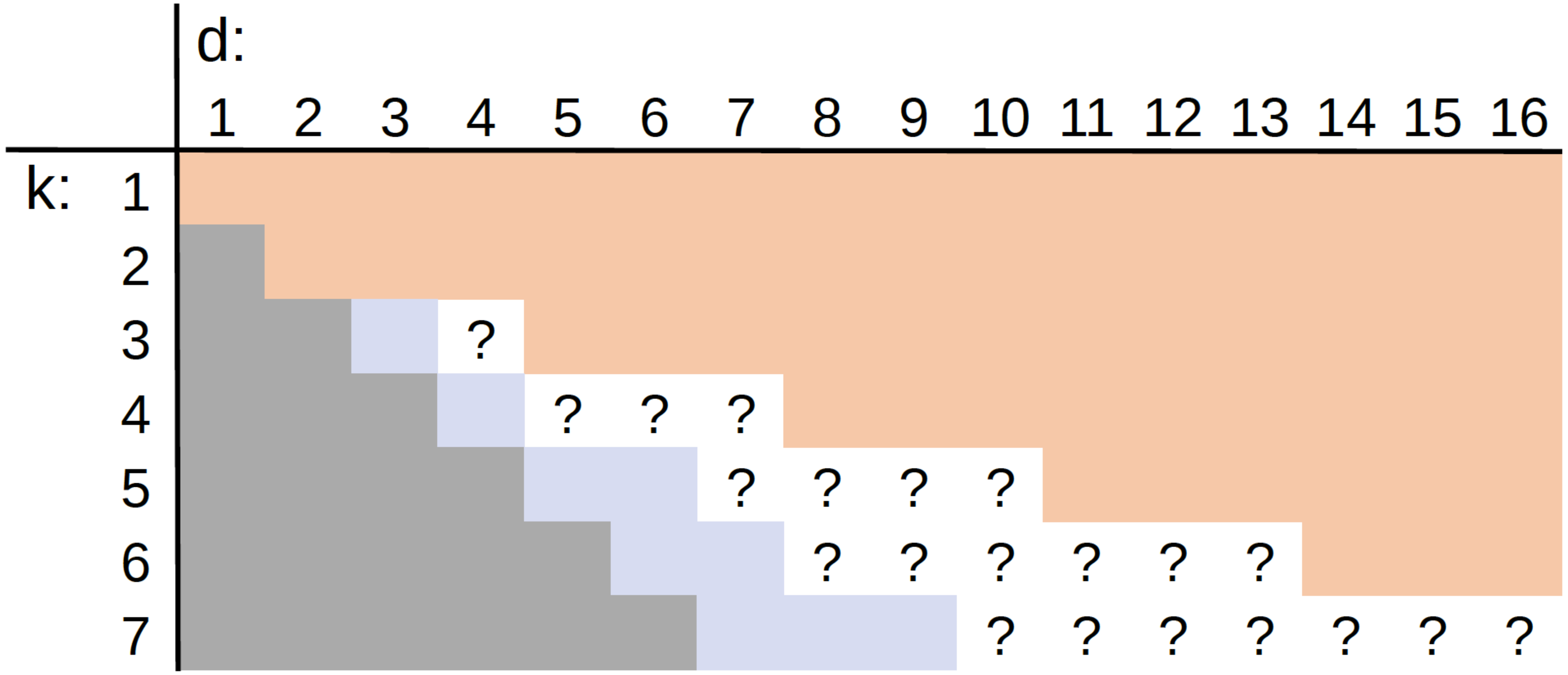
Summary

k*-partial 3-coloring in graphs with minimum degree *d

Easy

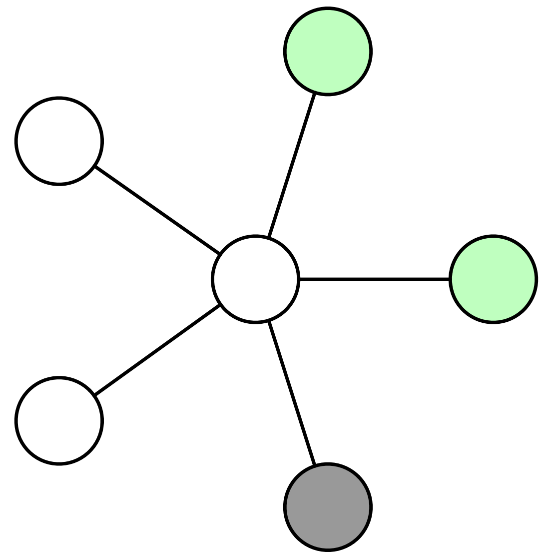
Hard

No solution

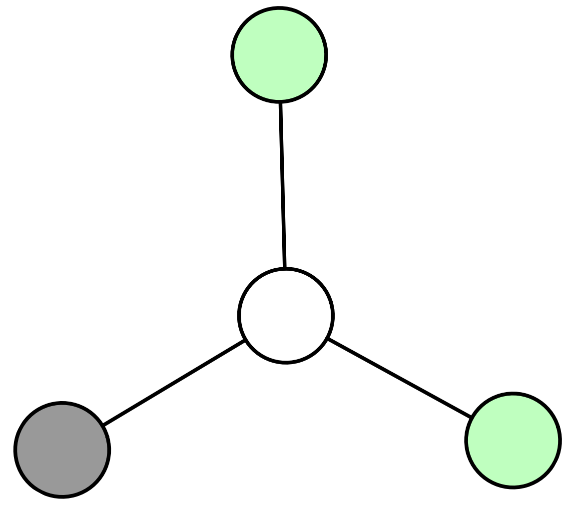


Concrete open problem

3-partial 3-coloring in **5**-regular trees **Easy**

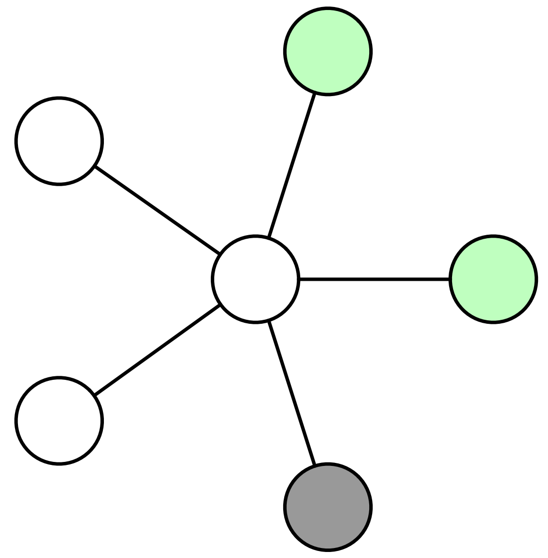


3-partial 3-coloring in **3**-regular trees **Hard**

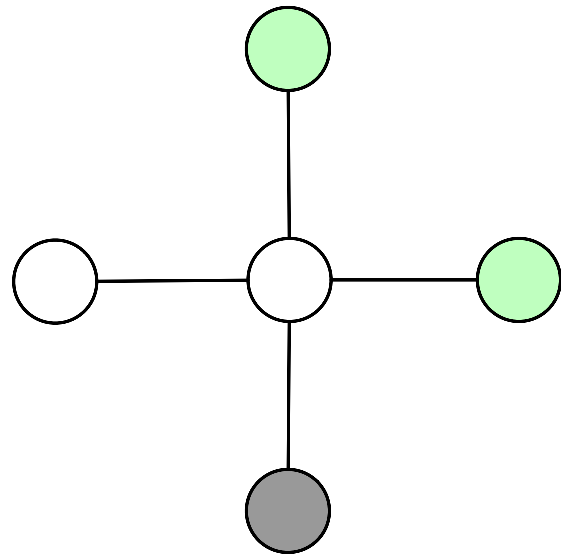


Concrete open problem

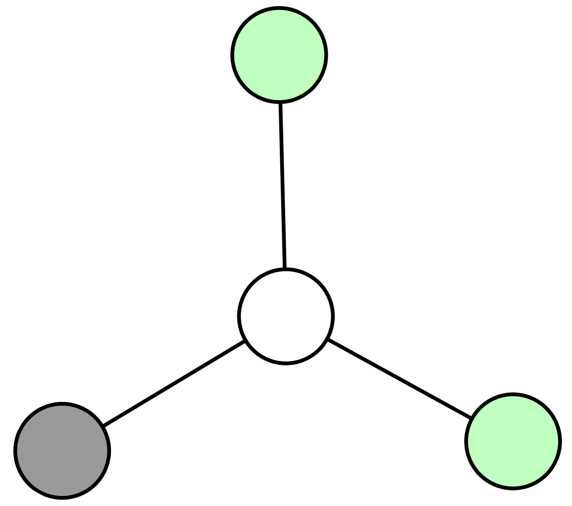
3-partial 3-coloring in **5**-regular trees **Easy**



3-partial 3-coloring in **4**-regular trees? **???**

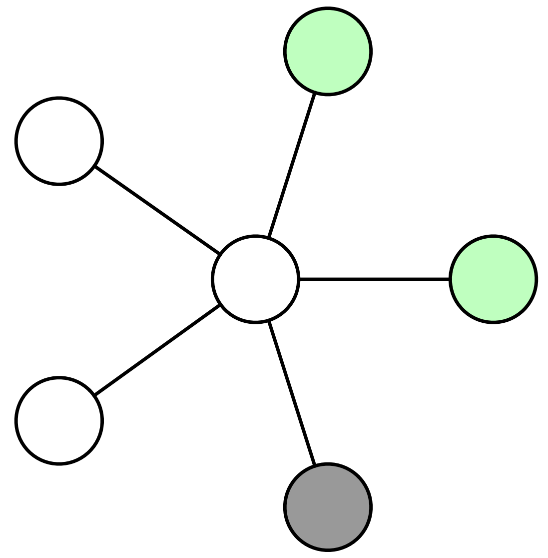


3-partial 3-coloring in **3**-regular trees **Hard**

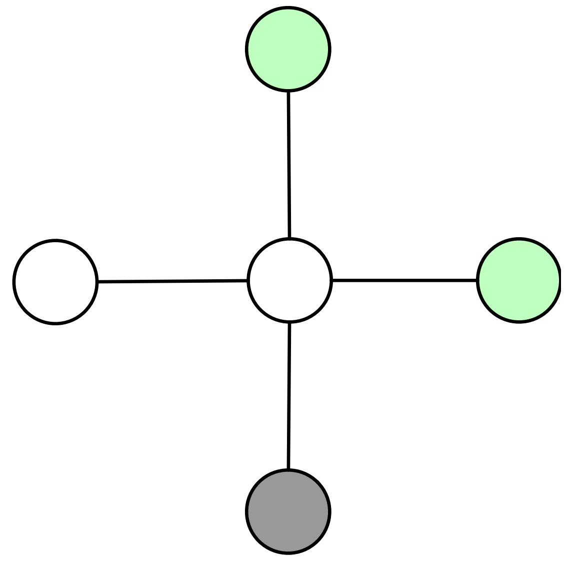


Concrete open problem

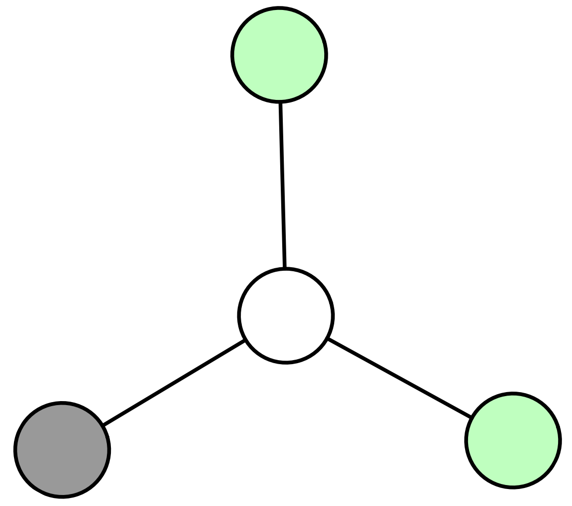
3-partial 3-coloring in **5**-regular trees — **Easy**



3-partial 3-coloring in **4**-regular trees? — **???**



3-partial 3-coloring in **3**-regular trees — **Hard**



Thank you!