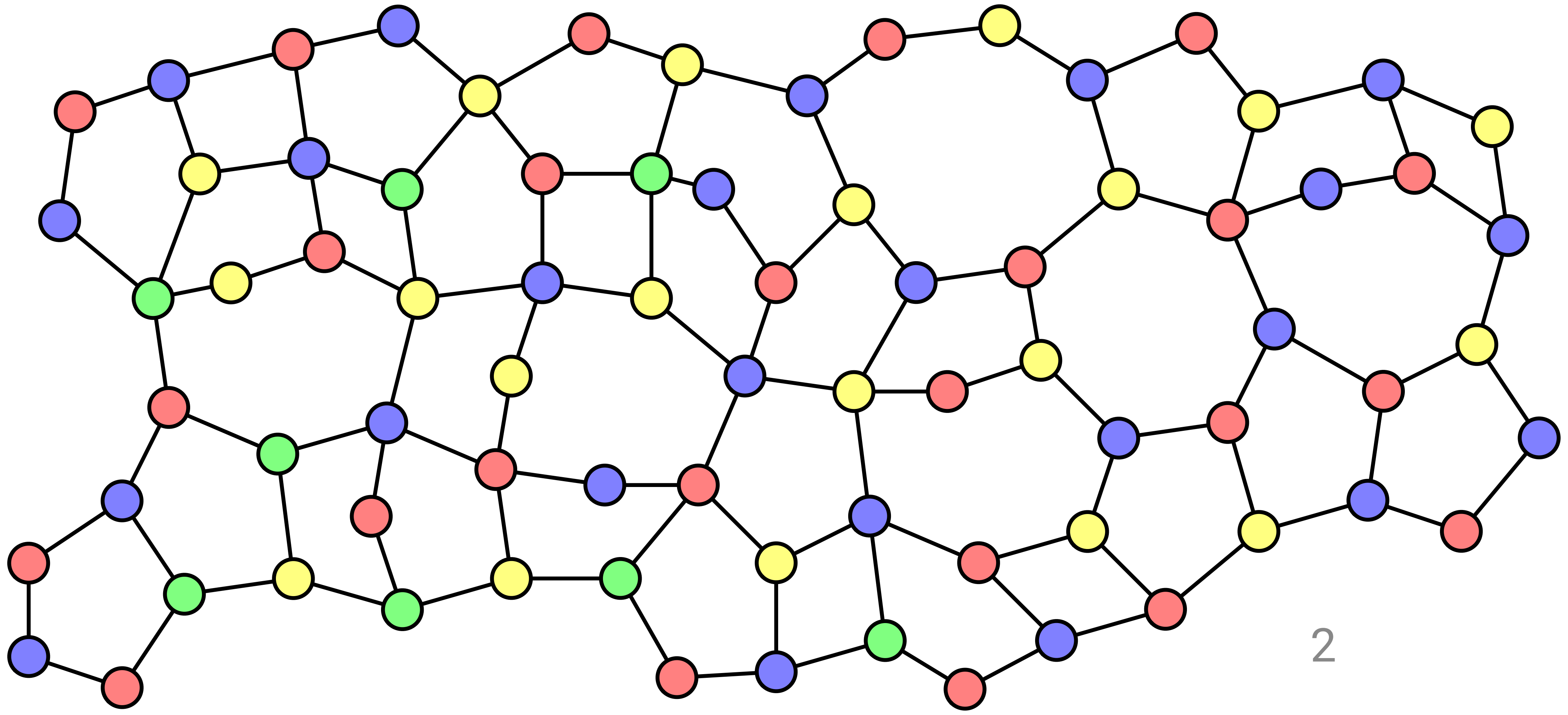
Randomized Coloring & MIS

Dennis Olivetti

University of Freiburg, Germany

Distributed Coloring Problem

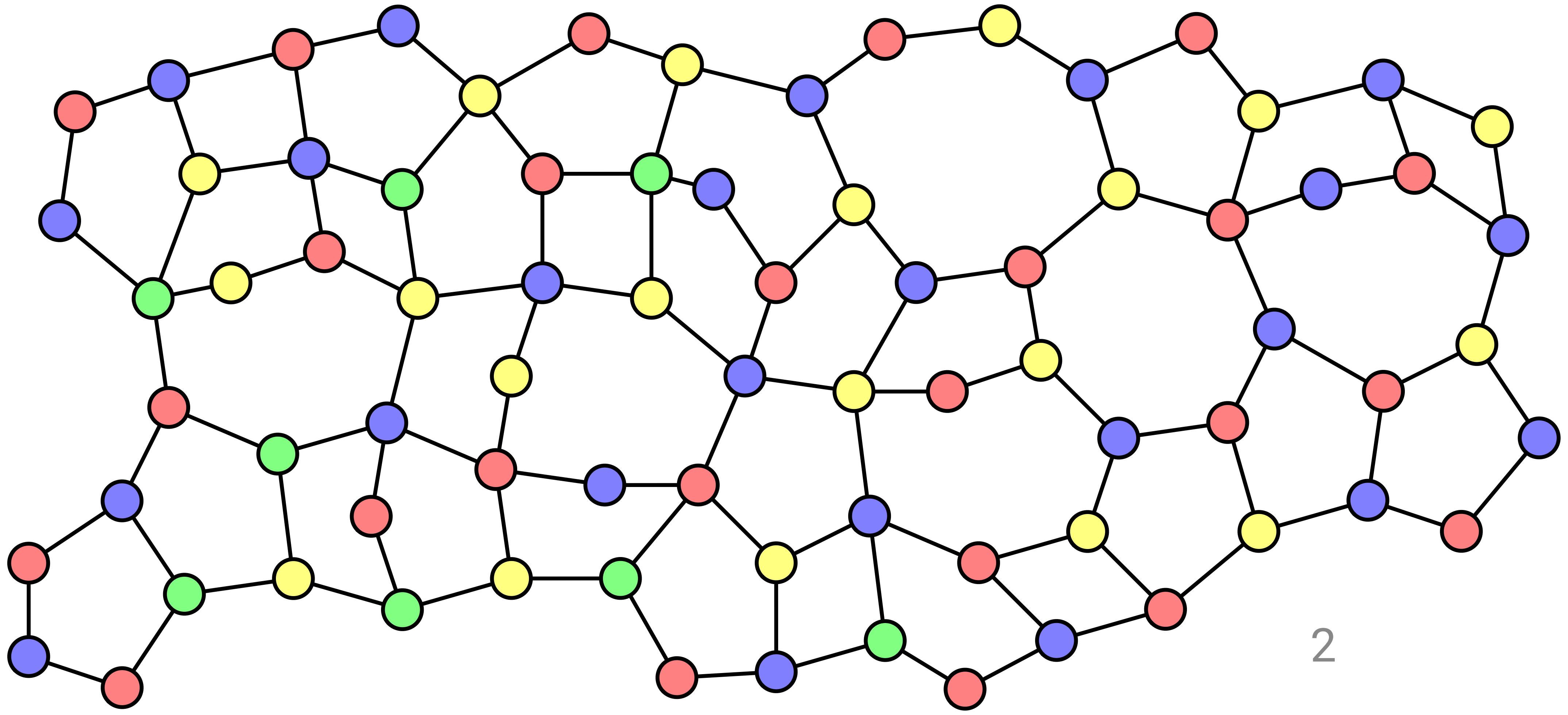
(Δ+1)-Vertex Coloring



Distributed Coloring Problem

Objective: properly color the nodes with **≤ Δ + 1** colors

(Δ+1)-Vertex Coloring

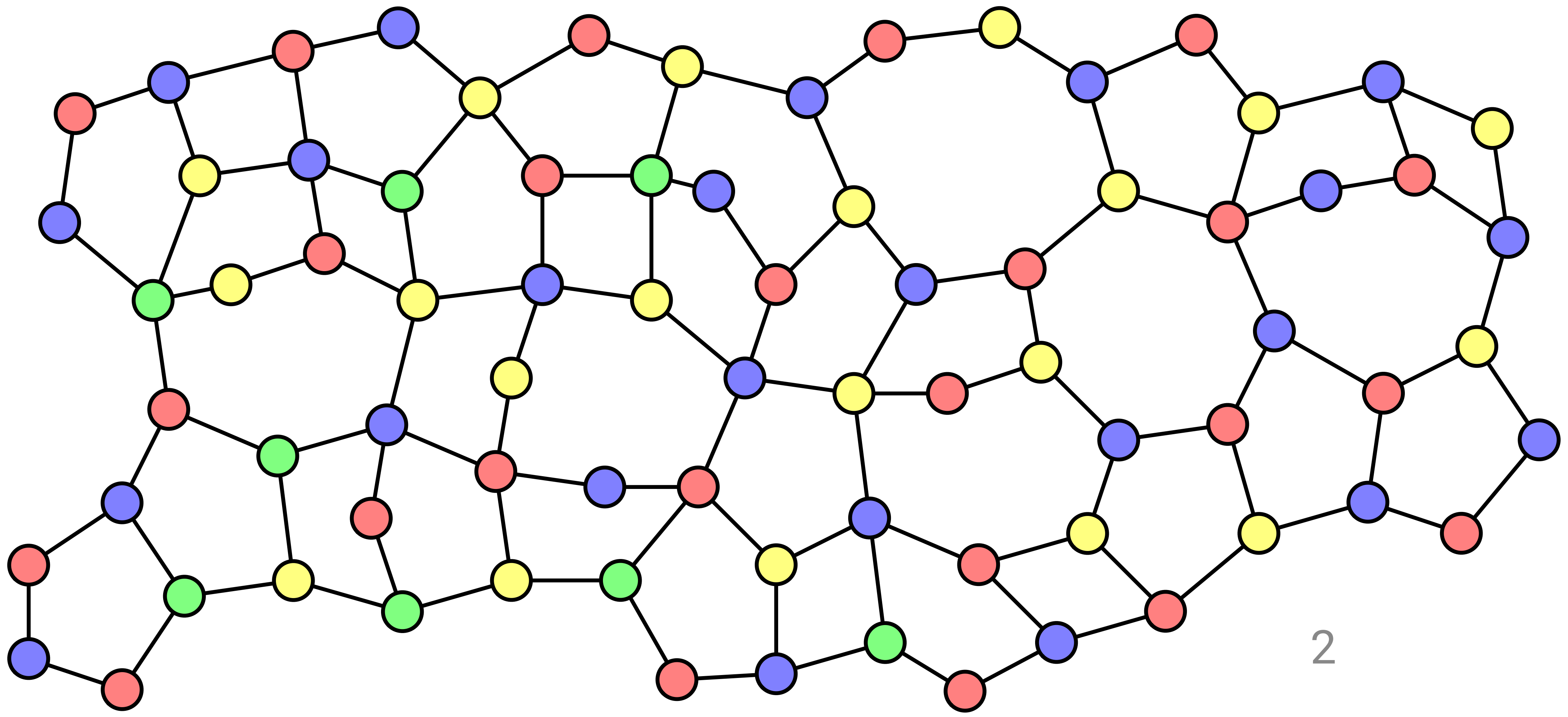


Distributed Coloring Problem

Objective: properly color the nodes with **≤ Δ + 1** colors

- **Δ:** maximum degree

(Δ+1)-Vertex Coloring

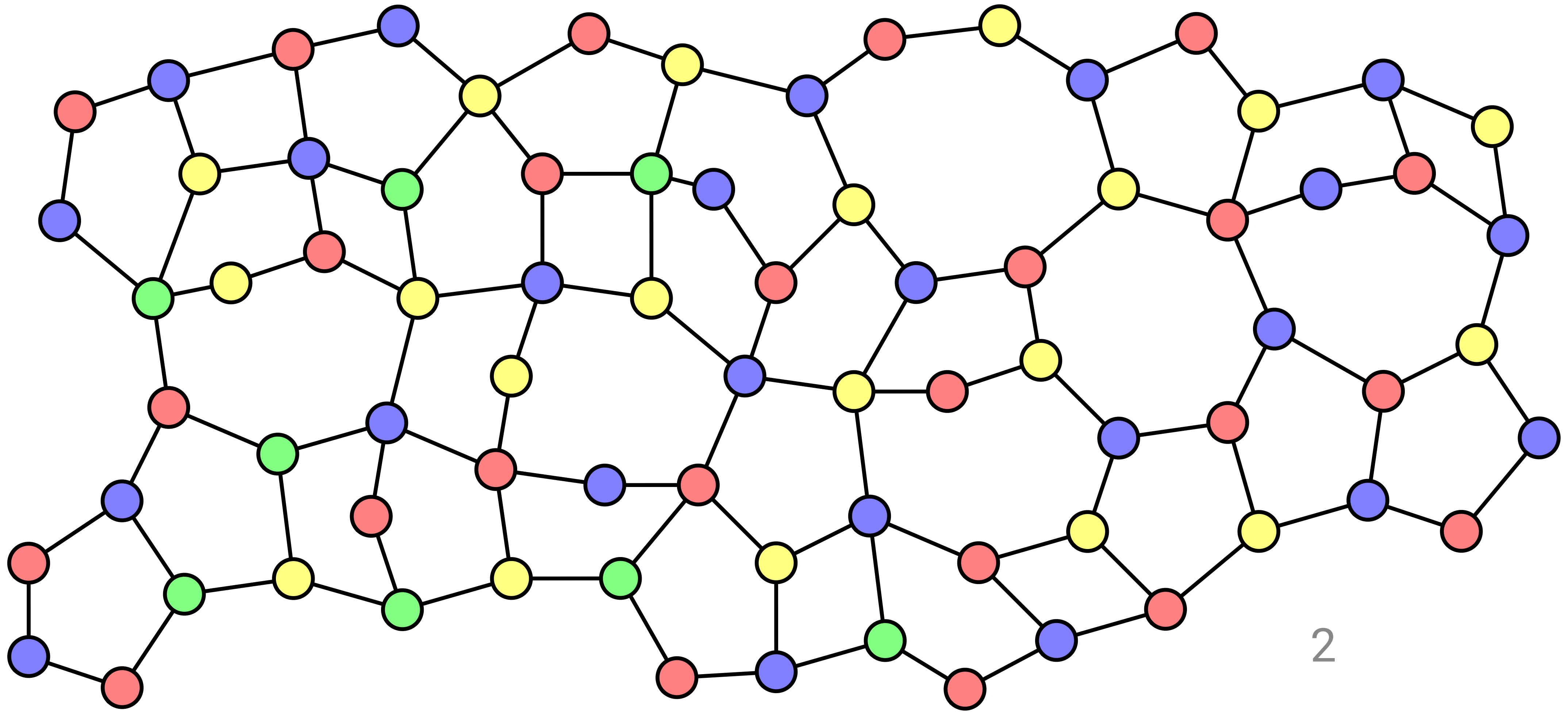


Distributed Coloring Problem

Objective: properly color the nodes with **≤ Δ + 1** colors

- **Δ:** maximum degree
- **Δ + 1:** what a simple sequential

(Δ+1)-Vertex Coloring

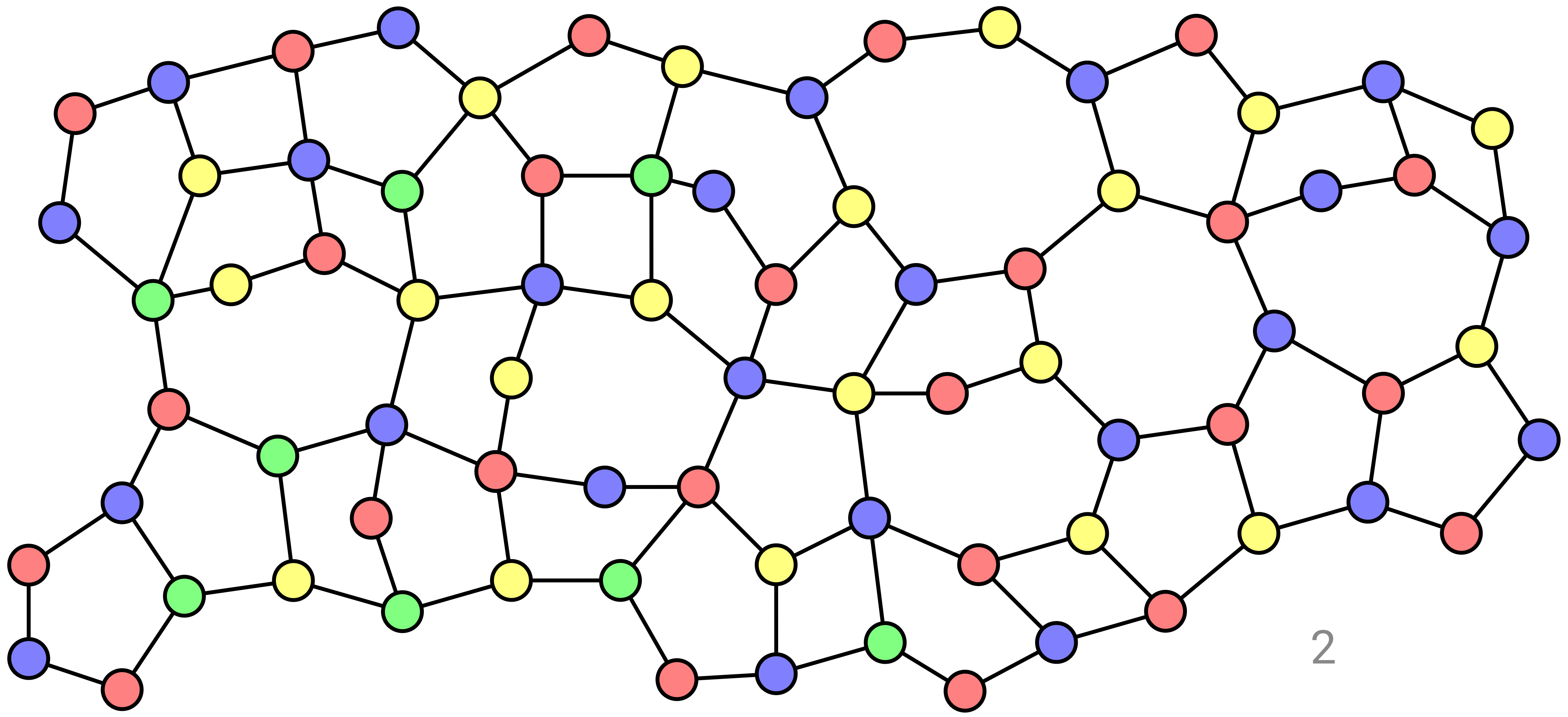


Distributed Coloring Problem

Objective: properly color the nodes with ≤ Δ + 1 colors

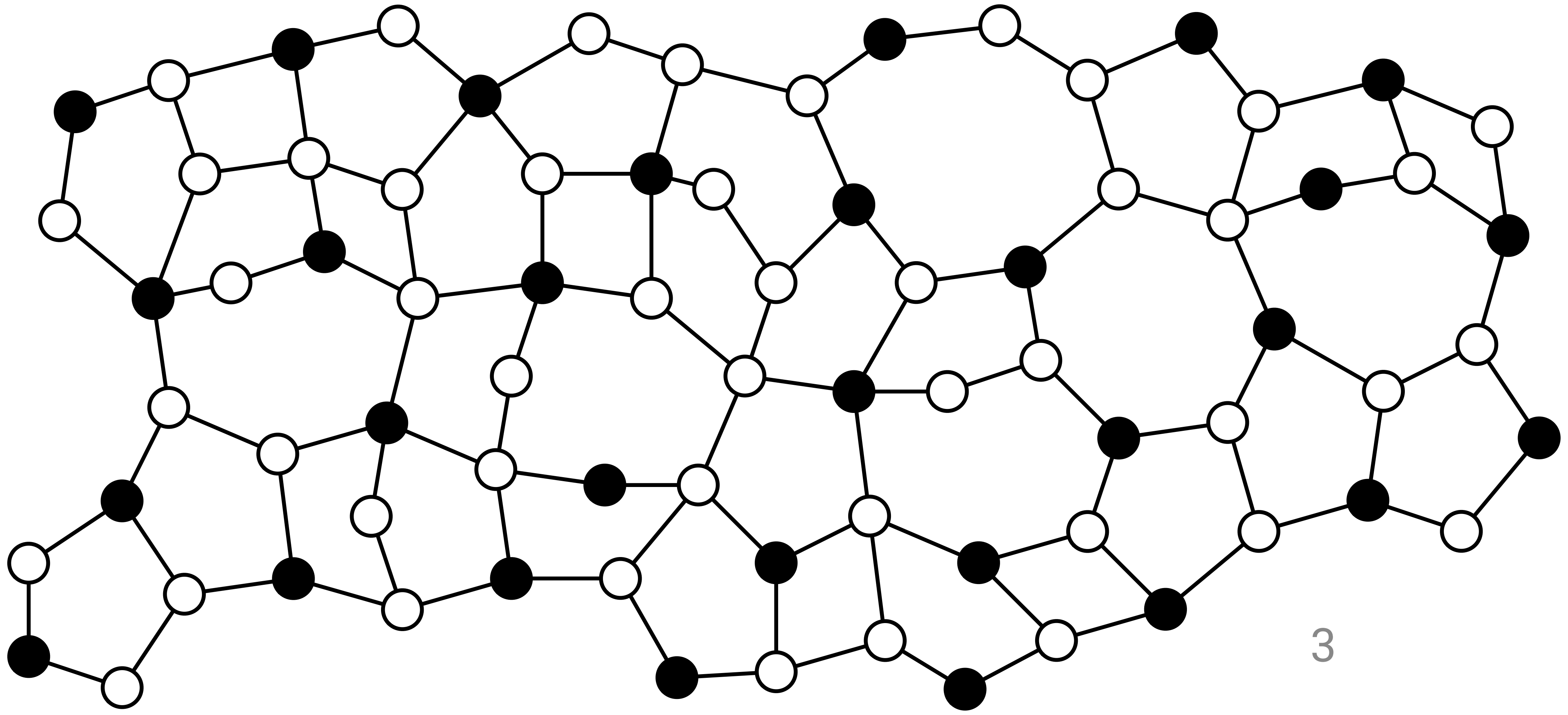
- Δ: maximum degree
- Δ + 1: what a simple sequential **greedy** algorithm achieves

(Δ+1)-Vertex Coloring



Distributed MIS Problem

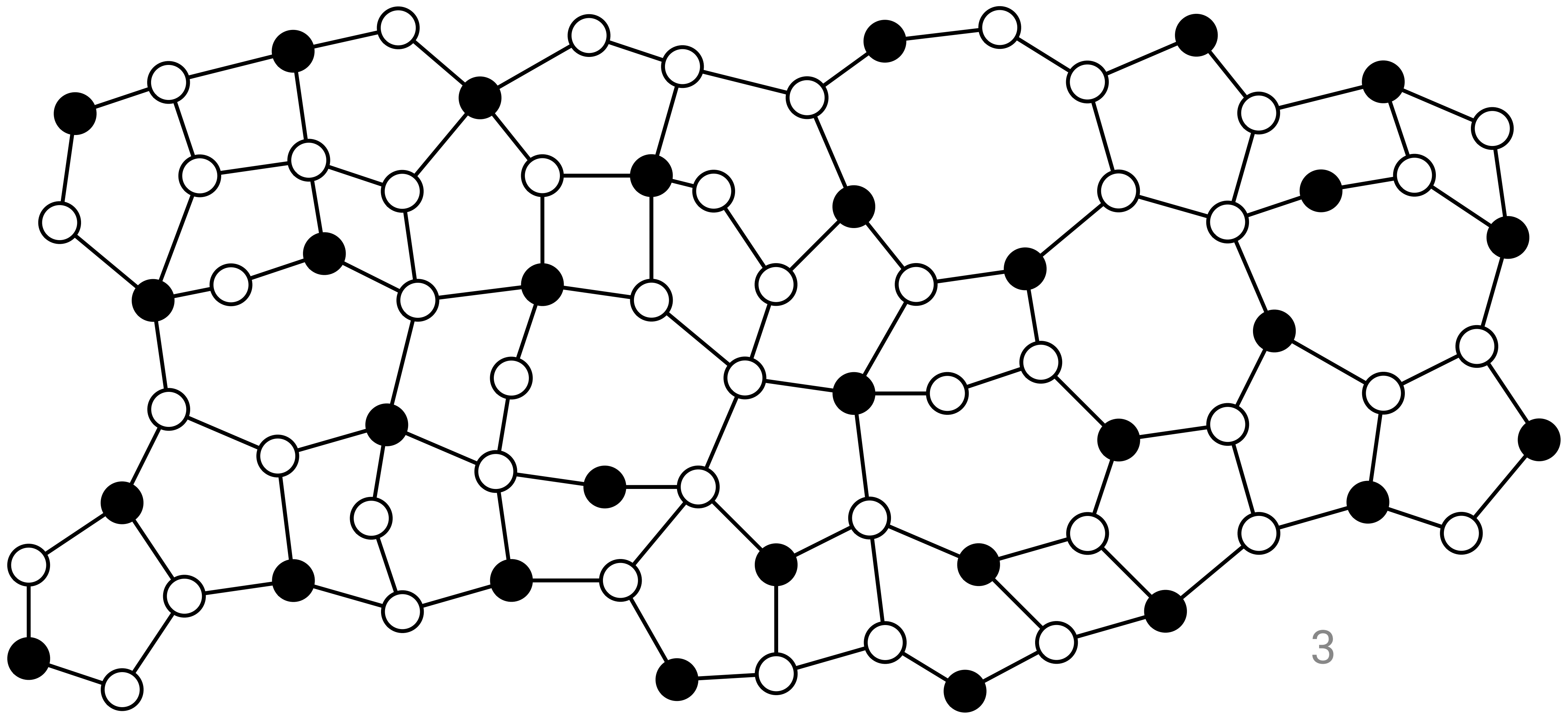
Maximal Independent Set



Distributed MIS Problem

Objective: compute a maximal independent set (MIS)

Maximal Independent Set

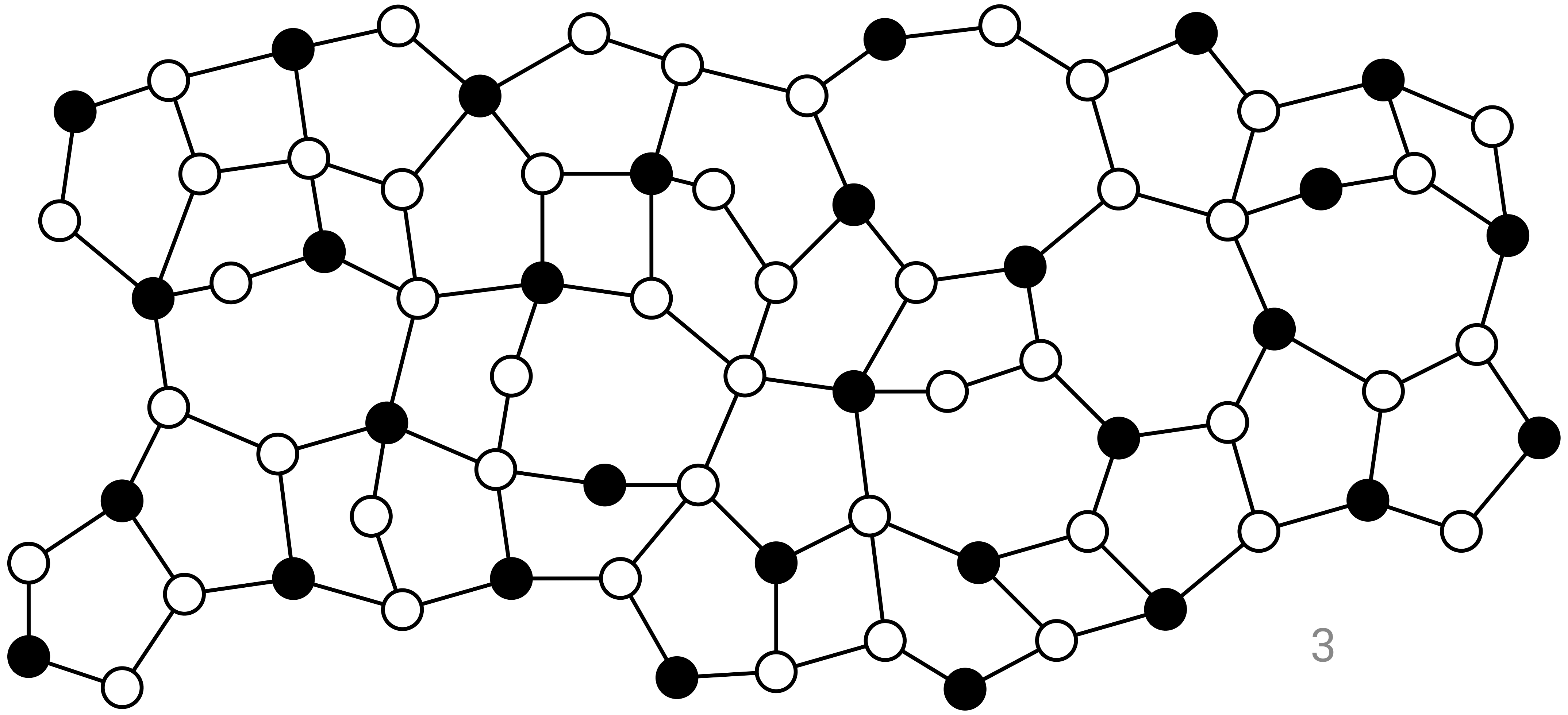


Distributed MIS Problem

Objective: compute a maximal independent set (MIS)

- **Independent Set:** set of pairwise non-adjacent nodes

Maximal Independent Set

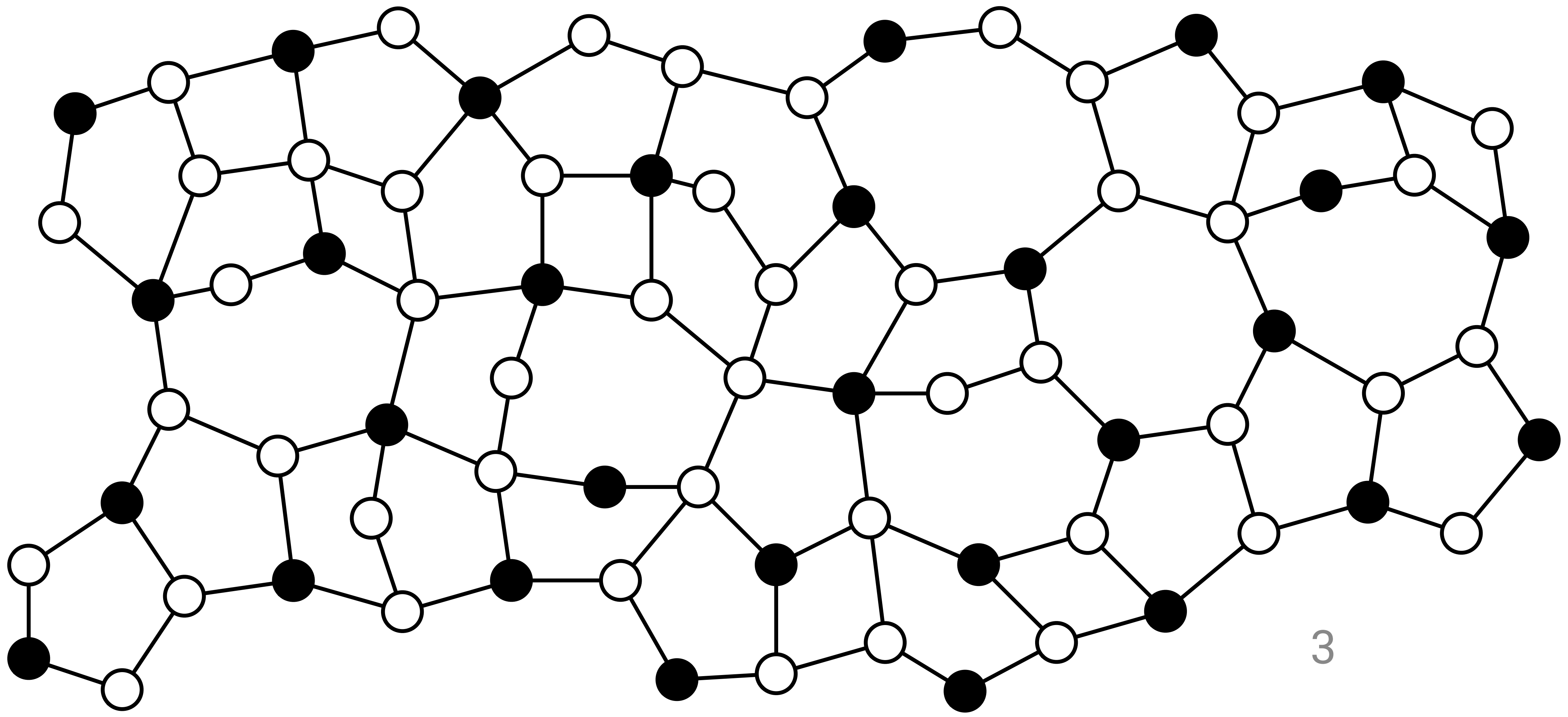


Distributed MIS Problem

Objective: compute a maximal independent set (MIS)

- **Independent Set:** set of pairwise non-adjacent nodes
- **Maximal:** the set cannot be extended

Maximal Independent Set

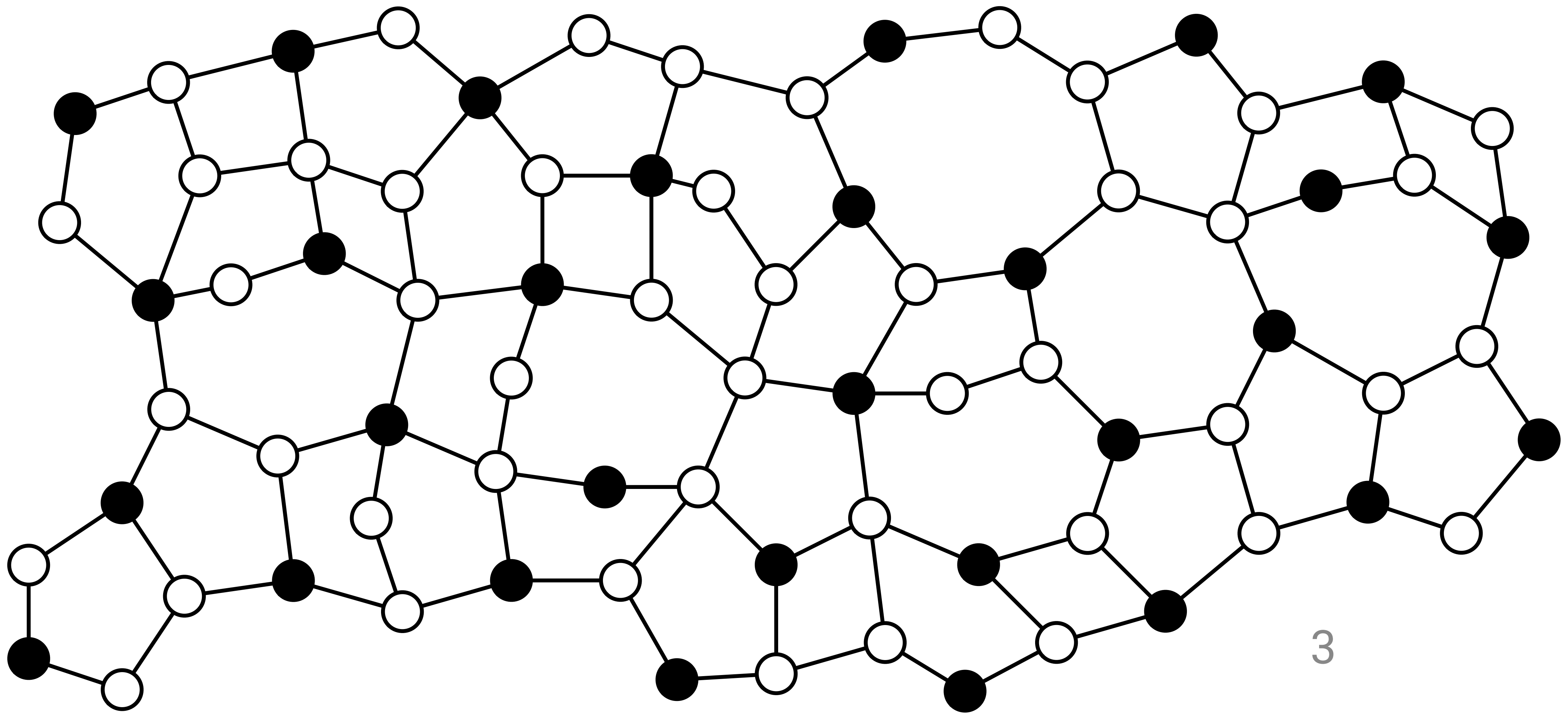


Distributed MIS Problem

Objective: compute a maximal independent set (MIS)

- **Independent Set:** set of pairwise non-adjacent nodes
- **Maximal:** the set cannot be extended
- Easily solvable with a **greedy** algorithm

Maximal Independent Set

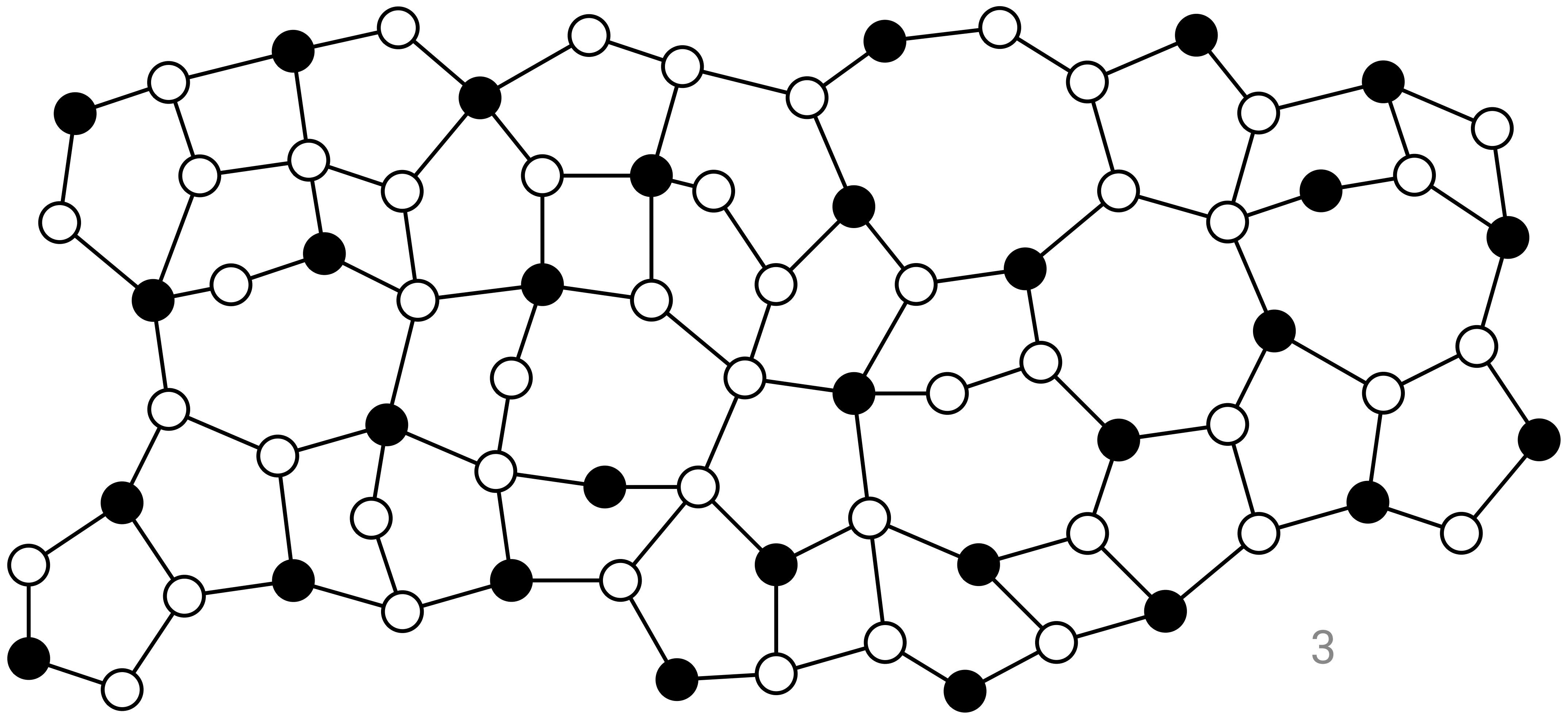


Distributed MIS Problem

Objective: compute a maximal independent set (MIS)

- **Independent Set:** set of pairwise non-adjacent nodes
- **Maximal:** the set cannot be extended
- Easily solvable with a **greedy** algorithm
- The Maximum Independent Set is a

Maximal Independent Set

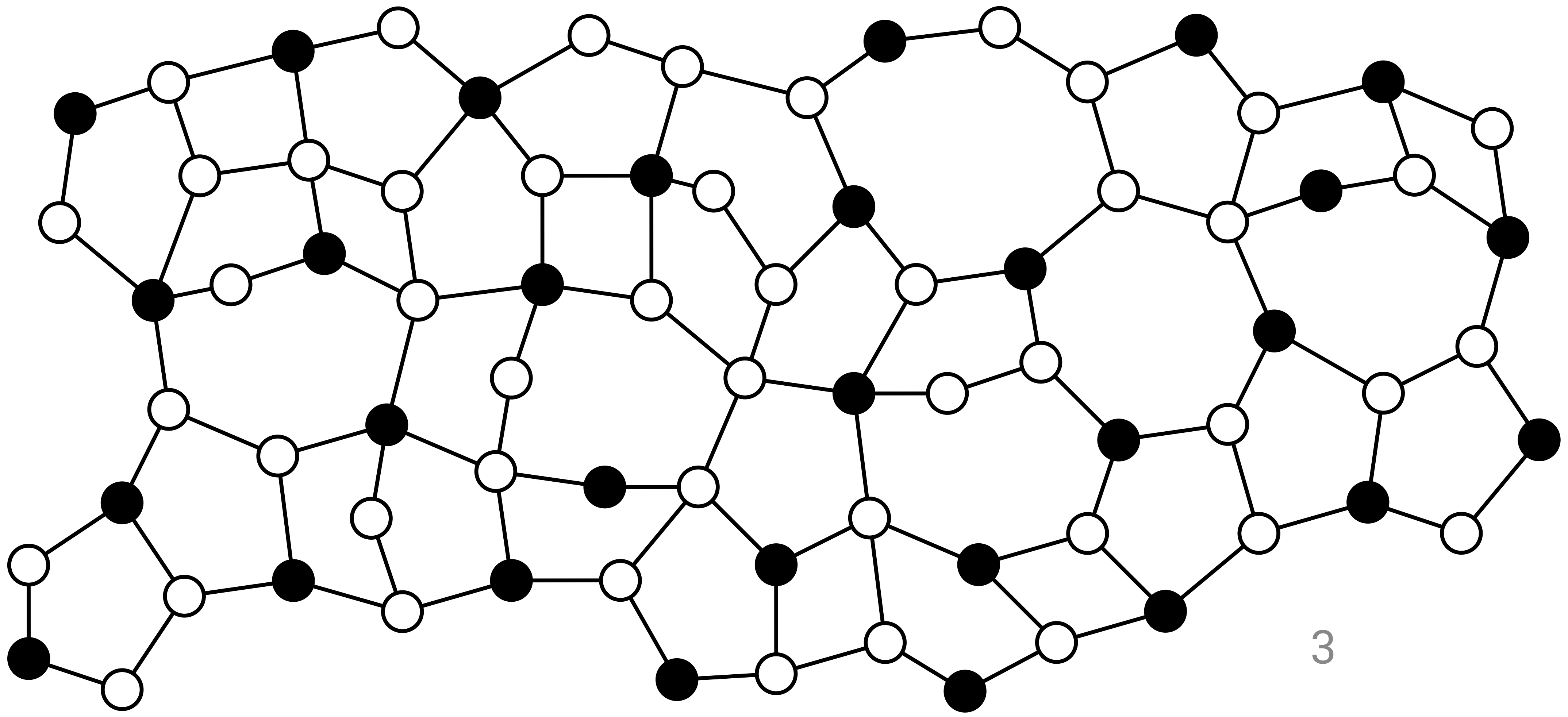


Distributed MIS Problem

Objective: compute a maximal independent set (MIS)

- **Independent Set:** set of pairwise non-adjacent nodes
- **Maximal:** the set cannot be extended
- Easily solvable with a **greedy** algorithm
- The Maximum Independent Set is a different (much harder) problem

Maximal Independent Set



Summary From Last Time

Summary From Last Time

- **Coloring trees**

Summary From Last Time

- **Coloring trees**
  - 2-coloring is solvable in trees, but requires a lot of time ($\Omega(D)$)

Summary From Last Time

- **Coloring trees**
  - 2-coloring is solvable in trees, but requires a lot of time (**$\Omega(D)$**)
  - 3-coloring in **rooted** trees can be solved in **$O(\log^* n)$** time

Summary From Last Time

- **Coloring trees**
  - 2-coloring is solvable in trees, but requires a lot of time (**Ω(D)**)
  - 3-coloring in **rooted** trees can be solved in **O(log* n)** time
  - 3-coloring in **unrooted** trees can be solved in **O(log n)** time

Summary From Last Time

- **Coloring trees**
  - 2-coloring is solvable in trees, but requires a lot of time (**Ω(D)**)
  - 3-coloring in **rooted** trees can be solved in **O(log* n)** time
  - 3-coloring in **unrooted** trees can be solved in **O(log n)** time
- **Coloring general graphs**

Summary From Last Time

- **Coloring trees**
  - 2-coloring is solvable in trees, but requires a lot of time ($\Omega(D)$)
  - 3-coloring in **rooted** trees can be solved in $O(\log^* n)$ time
  - 3-coloring in **unrooted** trees can be solved in $O(\log n)$ time
- **Coloring general graphs**
  - 3^{Δ}-coloring in graphs with max degree Δ can be solved in $O(\log^* n)$ time

Summary From Last Time

- **Coloring trees**
  - 2-coloring is solvable in trees, but requires a lot of time ($\Omega(D)$)
  - 3-coloring in **rooted** trees can be solved in $O(\log^* n)$ time
  - 3-coloring in **unrooted** trees can be solved in $O(\log n)$ time
- **Coloring general graphs**
  - 3^Δ-coloring in graphs with max degree Δ can be solved in $O(\log^* n)$ time
  - $(\Delta+1)$-coloring or **MIS** in graphs with max degree Δ can be solved in $O(3^\Delta + \log^* n)$ time

Summary From Last Time

- **Coloring trees**
  - 2-coloring is solvable in trees, but requires a lot of time ($\Omega(D)$)
  - 3-coloring in **rooted** trees can be solved in $O(\log^* n)$ time
  - 3-coloring in **unrooted** trees can be solved in $O(\log n)$ time
- **Coloring general graphs**
  - 3^Δ-coloring in graphs with max degree Δ can be solved in $O(\log^* n)$ time
  - $(\Delta+1)$-coloring or MIS in graphs with max degree Δ can be solved in $O(3^\Delta + \log^* n)$ time
    - Fast if $\Delta = O(1)$

Summary From Last Time

- **Coloring trees**
  - 2-coloring is solvable in trees, but requires a lot of time (**$\Omega(D)$**)
  - 3-coloring in **rooted** trees can be solved in **$O(\log^* n)$** time
  - 3-coloring in **unrooted** trees can be solved in **$O(\log n)$** time
- **Coloring general graphs**
  - 3^Δ-coloring in graphs with max degree **Δ** can be solved in **$O(\log^* n)$** time
  - **$(\Delta+1)$**-coloring or **MIS** in graphs with max degree **Δ** can be solved in **$O(3^\Delta + \log^* n)$** time
    - Fast if **$\Delta = O(1)$**
    - It can be significantly improved

Summary From Last Time

- **Coloring trees**
  - 2-coloring is solvable in trees, but requires a lot of time (**Ω(D)**)
  - 3-coloring in **rooted** trees can be solved in **O(log* n)** time
  - 3-coloring in **unrooted** trees can be solved in **O(log n)** time
- **Coloring general graphs**
  - 3^Δ-coloring in graphs with max degree **Δ** can be solved in **O(log* n)** time
  - **(Δ+1)**-coloring or **MIS** in graphs with max degree **Δ** can be solved in **$O(3^\Delta$ + log* n)** time
    - Fast if **Δ = O(1)**
    - It can be significantly improved
- **Today**

Summary From Last Time

- **Coloring trees**
  - 2-coloring is solvable in trees, but requires a lot of time ($\Omega(D)$)
  - 3-coloring in **rooted** trees can be solved in $O(\log^* n)$ time
  - 3-coloring in **unrooted** trees can be solved in $O(\log n)$ time
- **Coloring general graphs**
  - 3^Δ-coloring in graphs with max degree Δ can be solved in $O(\log^* n)$ time
  - $(\Delta+1)$-coloring or **MIS** in graphs with max degree Δ can be solved in $O(3^\Delta + \log^* n)$ time
    - Fast if $\Delta = O(1)$
    - It can be significantly improved
- **Today**
  - **Randomized algorithms** for $(\Delta+1)$-coloring and MIS: $O(\log n)$ time in general graphs!

Some Useful Inequalities

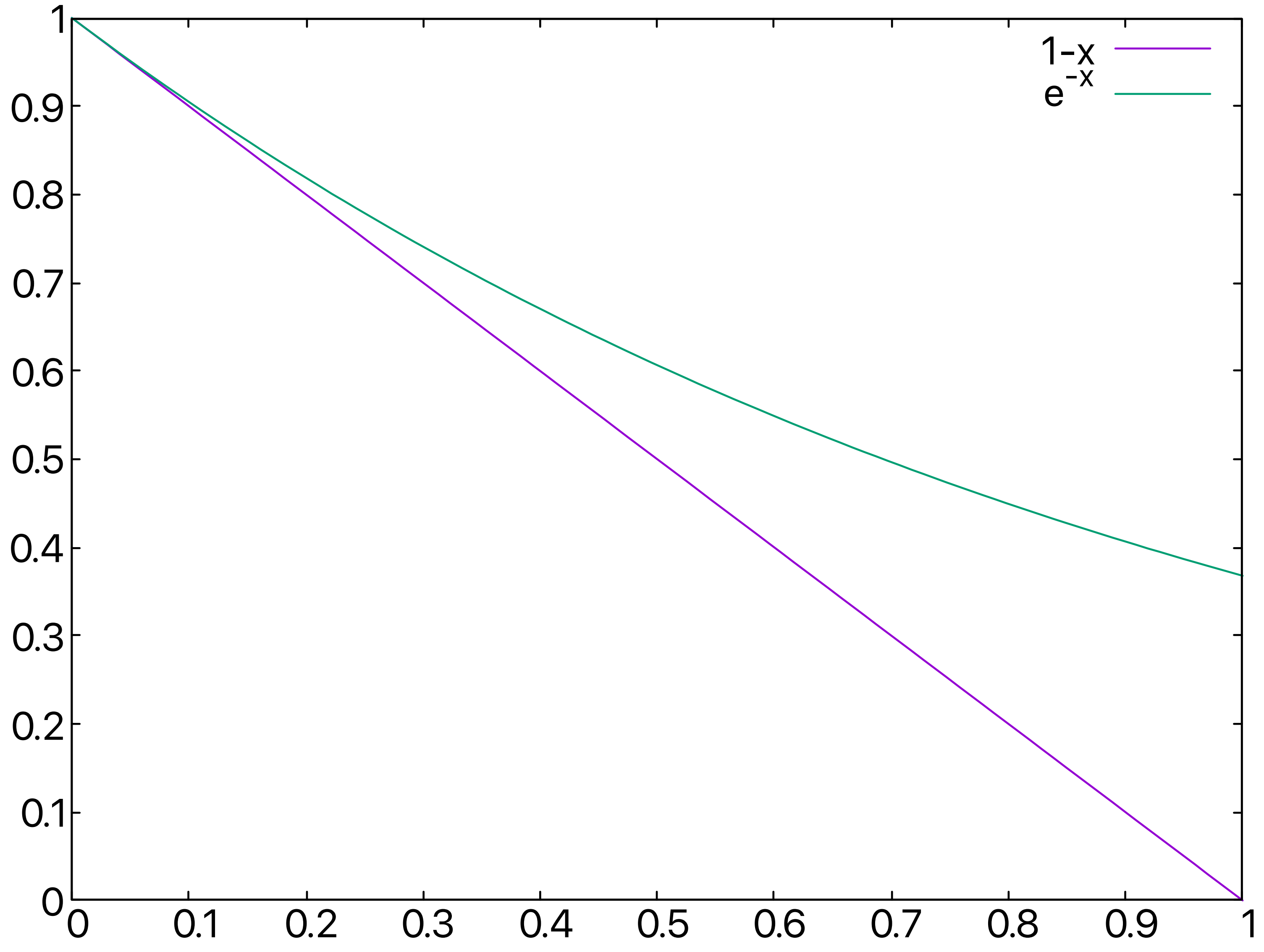
Some Useful Inequalities

- **Finding upper and lower bounds for $(1-x)^y$**

Some Useful Inequalities

- **Finding upper and lower bounds for $(1-x)^y$**

- $1 - x \leq e^{-x}$ for all $x \in \mathbb{R}$

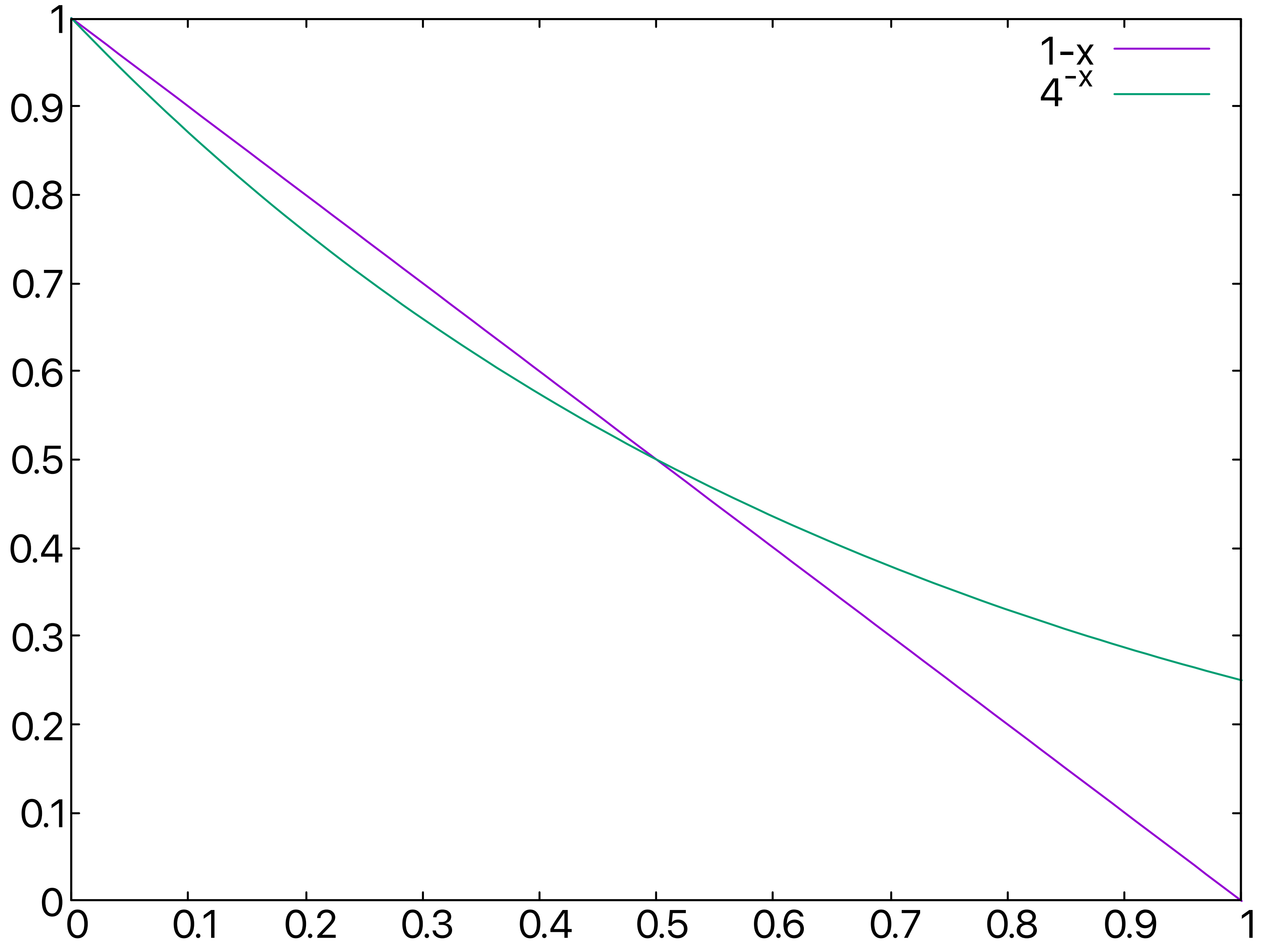


Some Useful Inequalities

- **Finding upper and lower bounds for $(1-x)^y$**

- $1 - x \leq e^{-x}$ for all $x \in \mathbb{R}$

- $1 - x \geq 4^{-x}$ for all $x \in [0, 1/2]$

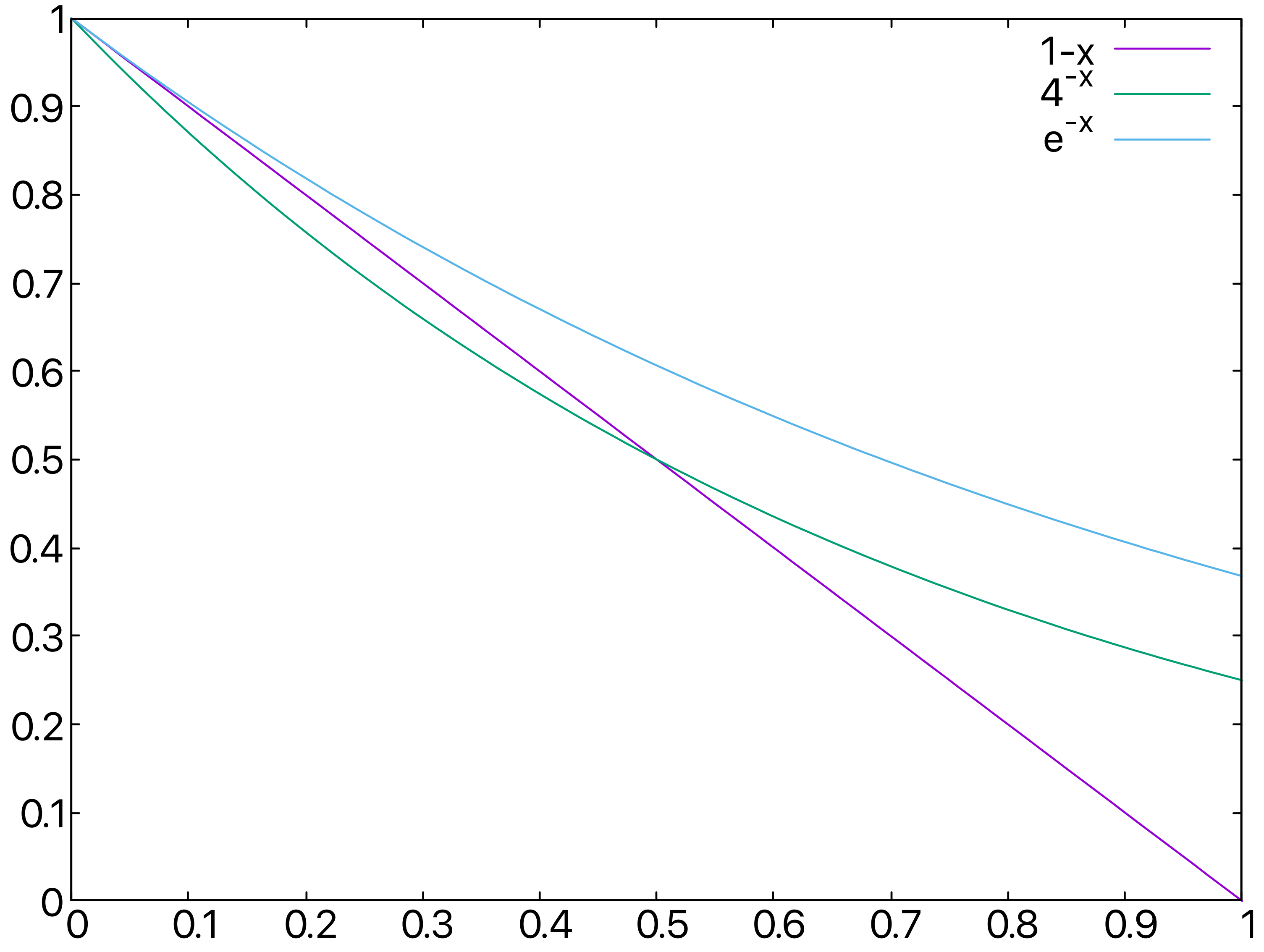


Some Useful Inequalities

- **Finding upper and lower bounds for $(1-x)^y$**

- $1 - x \leq e^{-x}$ for all $x \in \mathbb{R}$

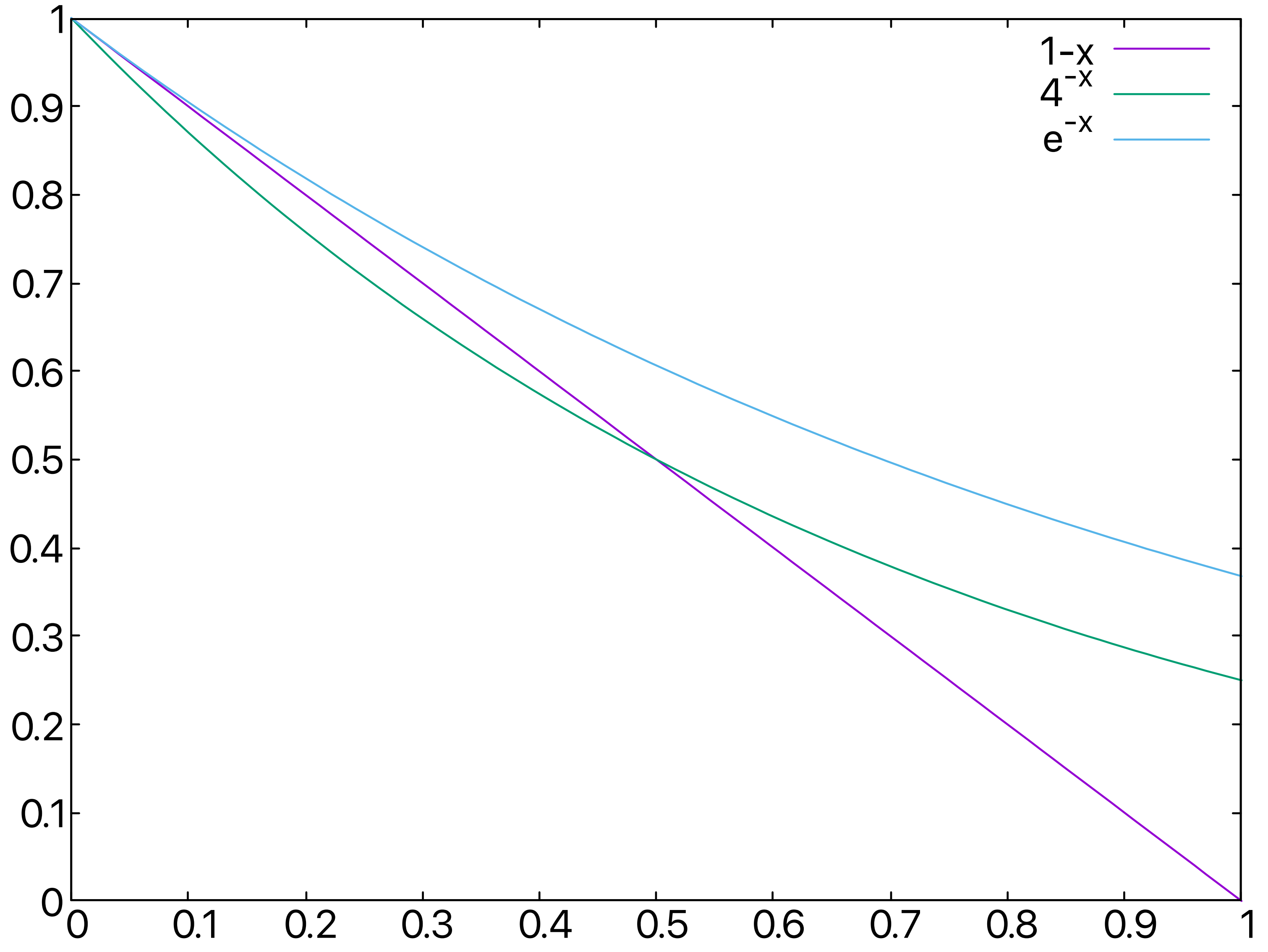
- $1 - x \geq 4^{-x}$ for all $x \in [0, 1/2]$
  - $4^{-x} \leq 1 - x \leq e^{-x}$ for all $x \in [0, 1/2]$

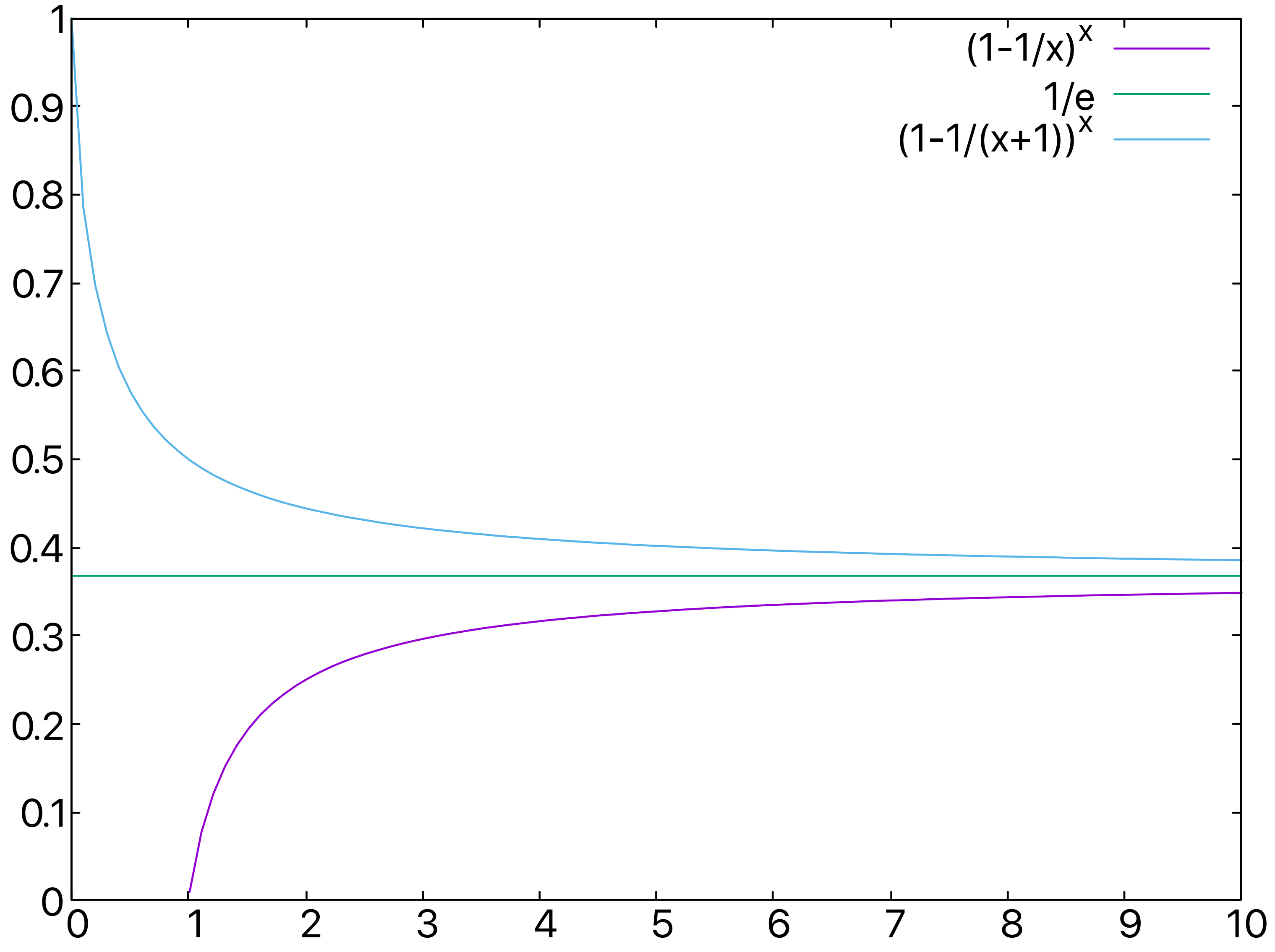


Some Useful Inequalities

- **Finding upper and lower bounds for $(1-x)^y$**

- $1 - x \leq e^{-x}$ for all $x \in \mathbb{R}$
- $1 - x \geq 4^{-x}$ for all $x \in [0, 1/2]$
    - $4^{-x} \leq 1 - x \leq e^{-x}$ for all $x \in [0, 1/2]$
- $\lim_{x\to\infty} (1 - 1/x)^x = 1/e$
    - $(1 - 1/x)^x < 1/e$ for all $x \geq 1$
    - $(1 - 1/(x+1))^x > 1/e$ for all $x > 0$





Randomized Coloring: Ideas

Randomized Coloring: Ideas

- **Problem:**

Randomized Coloring: Ideas

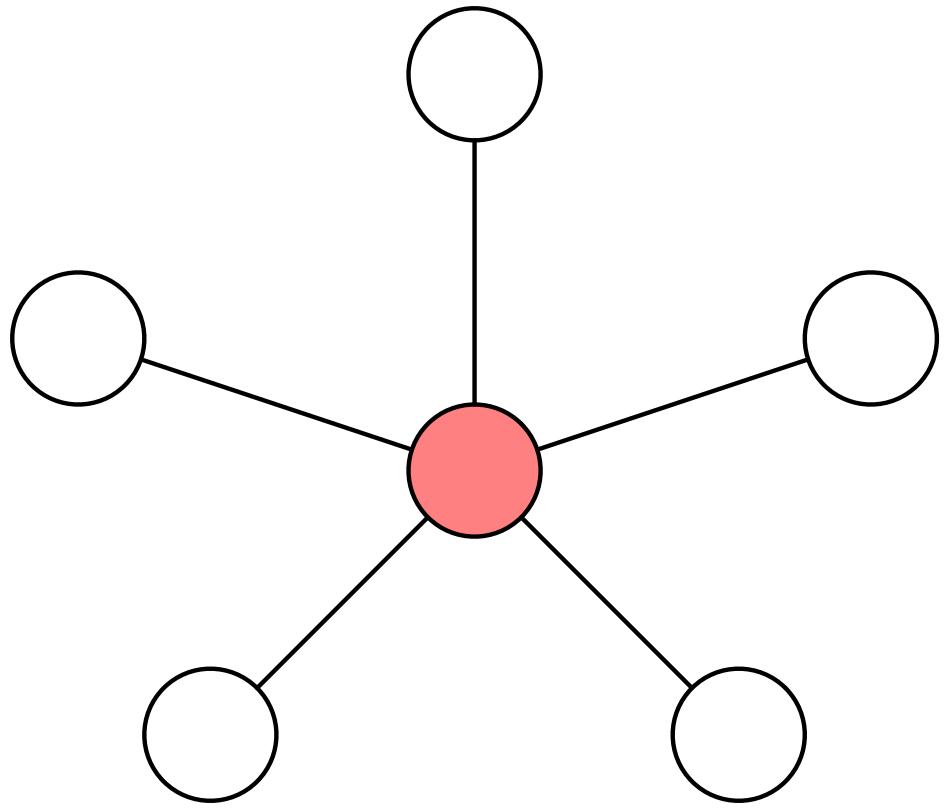
- **Problem:**
  - assign to each node a color from **$\{1, \ldots, \Delta + 1\}$**

Randomized Coloring: Ideas

- **Problem:**
  - assign to each node a color from **$\{1, \ldots, \Delta + 1\}$**
- **Simple idea:**

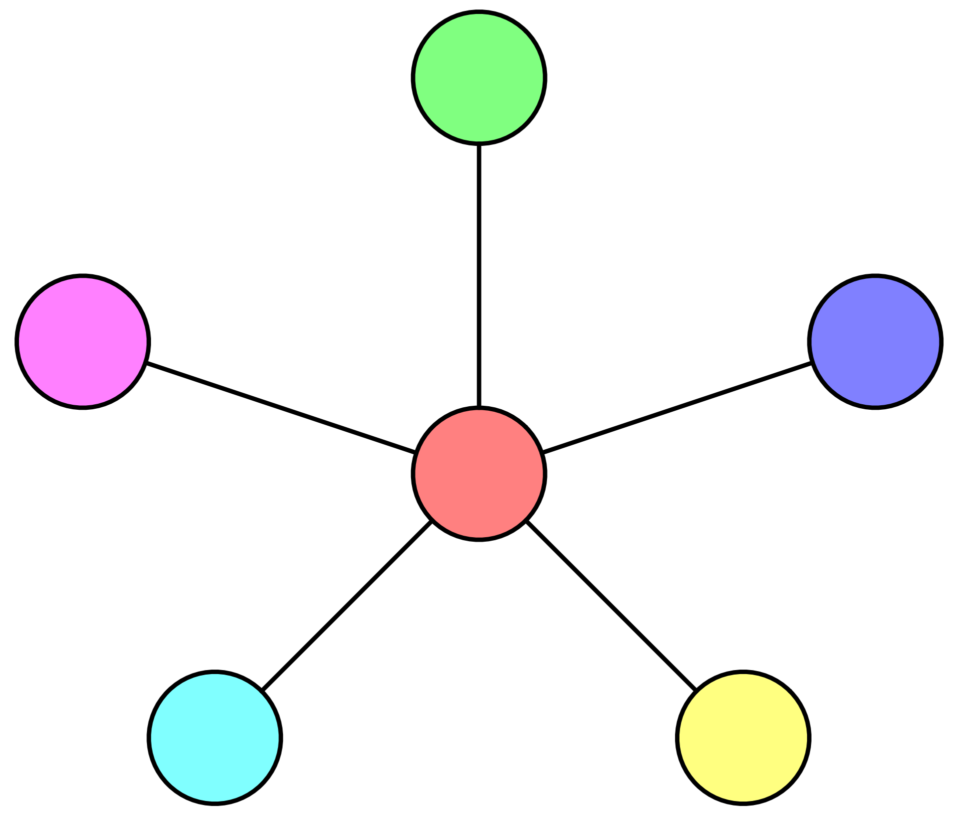
Randomized Coloring: Ideas

- **Problem:**
  - assign to each node a color from {1, ..., Δ + 1}
- **Simple idea:**
  - just pick a **random** color



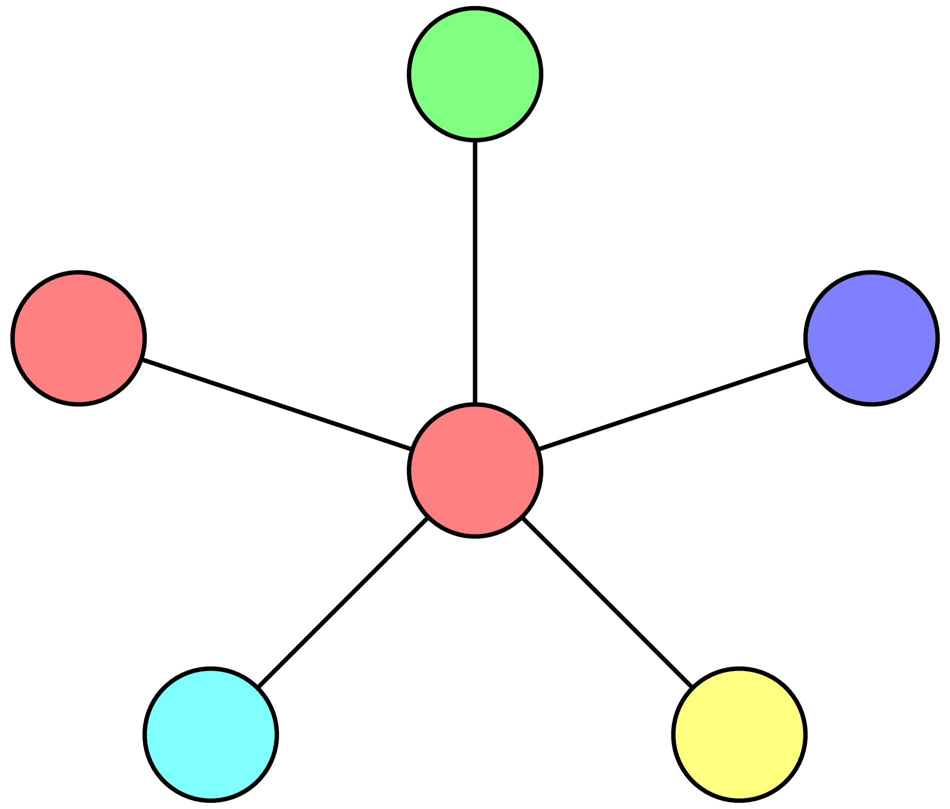
Randomized Coloring: Ideas

- **Problem:**
  - assign to each node a color from {1, ..., Δ + 1}
- **Simple idea:**
  - just pick a **random** color
  - if no neighbor picked the same color, **keep** the color



Randomized Coloring: Ideas

- **Problem:**
  - assign to each node a color from **{1, ..., Δ + 1}**
- **Simple idea:**
  - just pick a **random** color
  - if no neighbor picked the same color, **keep** the color
  - otherwise, **repeat**



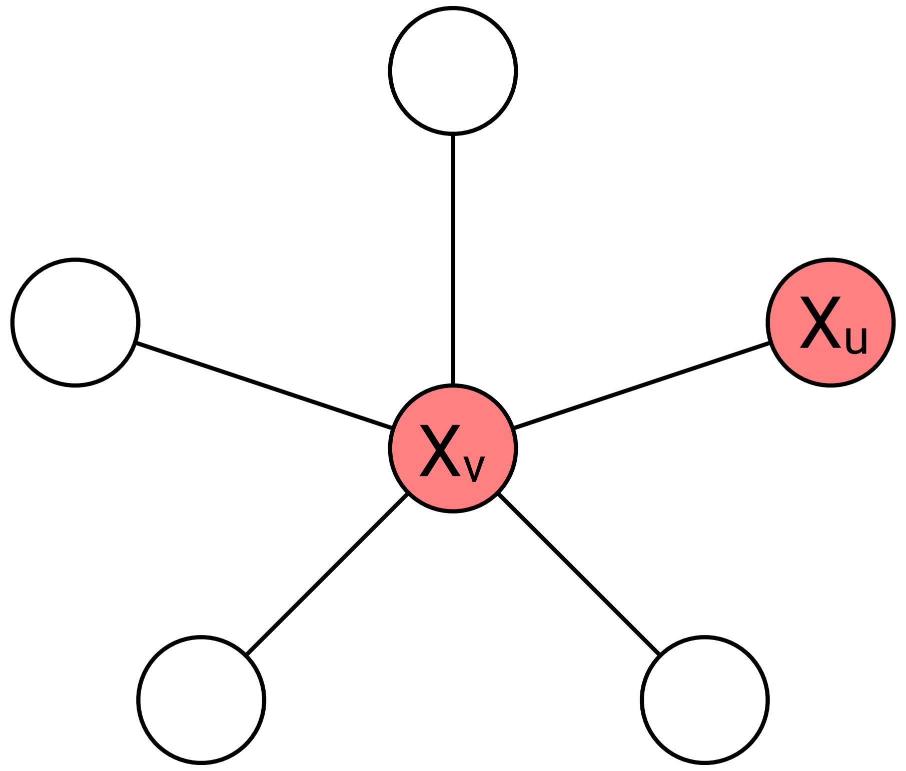
Random Colors

Random Colors

- **Lemma:** If each node **$v \in V$** of a graph **$G = (V,E)$** independently picks a **uniformly random** color **X_v** from **$\{1, \ldots, \Delta+1\}$**, for each node **$v \in V$**, the **probability** that **$X_v \neq X_u$** for all neighbors **u** of **v** is **at least $1/e$**.

Random Colors

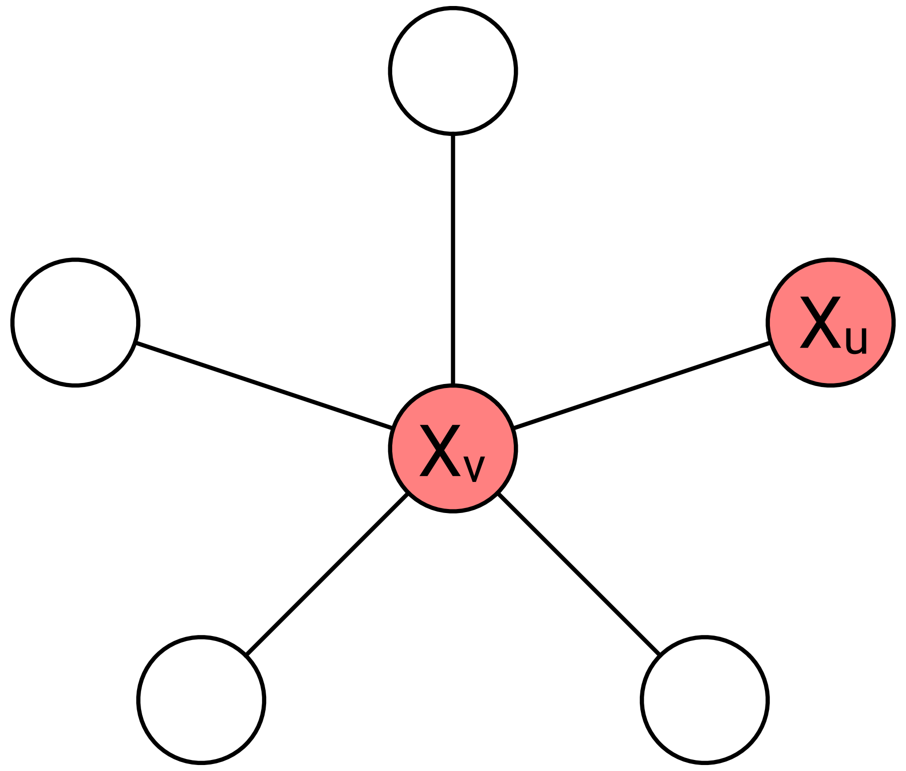
- **Lemma:** If each node $v \in V$ of a graph $G = (V,E)$ independently picks a **uniformly random** color X_v from $\{1, ..., \Delta+1\}$, for each node $v \in V$, the **probability** that $X_v \neq X_u$ for all neighbors u of v is **at least 1/e**.
- $P(X_v = X_u)$



Random Colors

- **Lemma:** If each node $v \in V$ of a graph $G = (V,E)$ independently picks a **uniformly random** color X_v from $\{1, \ldots, \Delta+1\}$, for each node $v \in V$, the **probability** that $X_v \neq X_u$ for all neighbors u of v is **at least 1/e**.
- $P(X_v = X_u)$

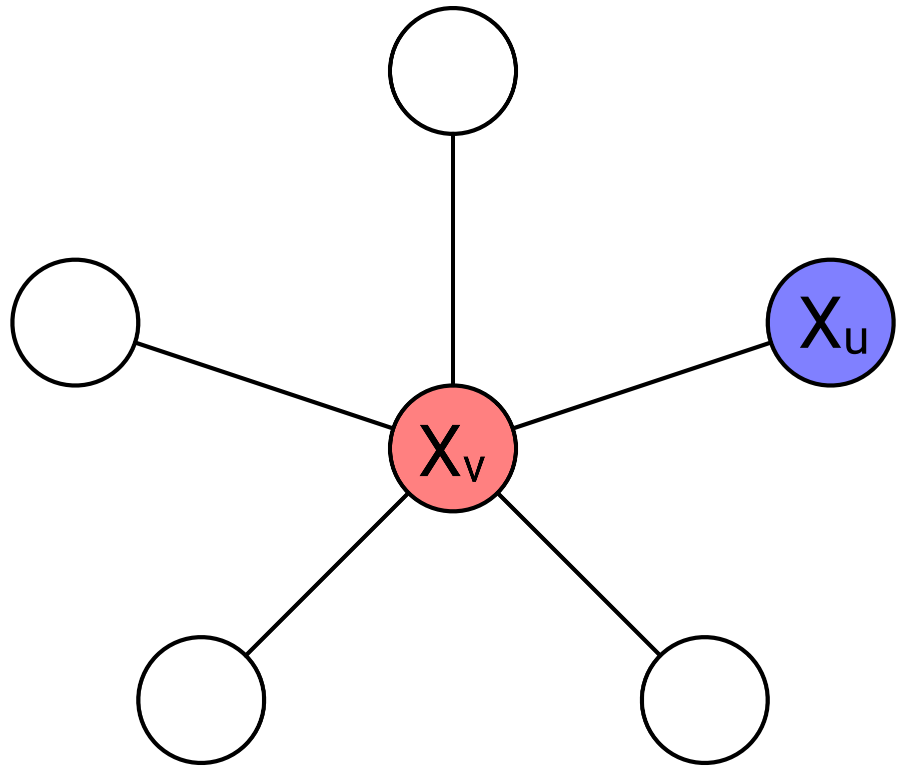
$$= 1 / (\Delta+1)$$



Random Colors

- **Lemma:** If each node $v \in V$ of a graph $G = (V,E)$ independently picks a **uniformly random** color X_v from $\{1, ..., \Delta+1\}$, for each node $v \in V$, the **probability** that $X_v \neq X_u$ for all neighbors u of v is **at least 1/e**.
- $P(X_v = X_u)$

  $= 1 / (\Delta+1)$
- $P(X_v \neq X_u)$



Random Colors

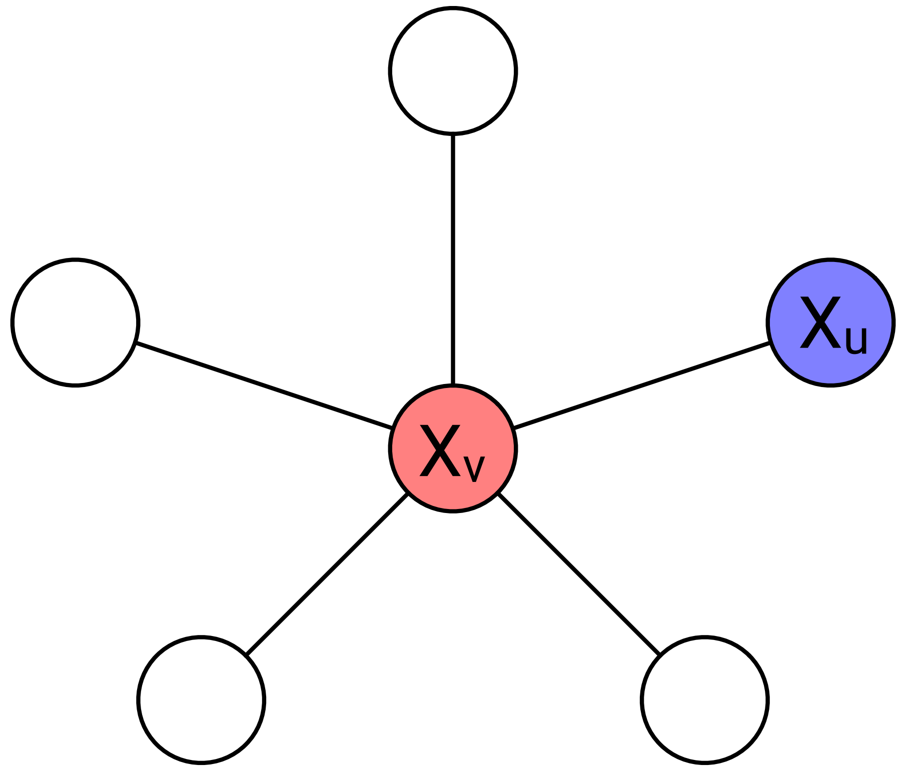
- **Lemma:** If each node $v \in V$ of a graph $G = (V,E)$ independently picks a **uniformly random** color X_v from $\{1, \ldots, \Delta+1\}$, for each node $v \in V$, the **probability** that $X_v \neq X_u$ for all neighbors u of v is **at least 1/e**.

- P($X_v = X_u$)

  $= 1 / (\Delta+1)$

- P($X_v \neq X_u$)

  $= 1 - 1 / (\Delta+1)$

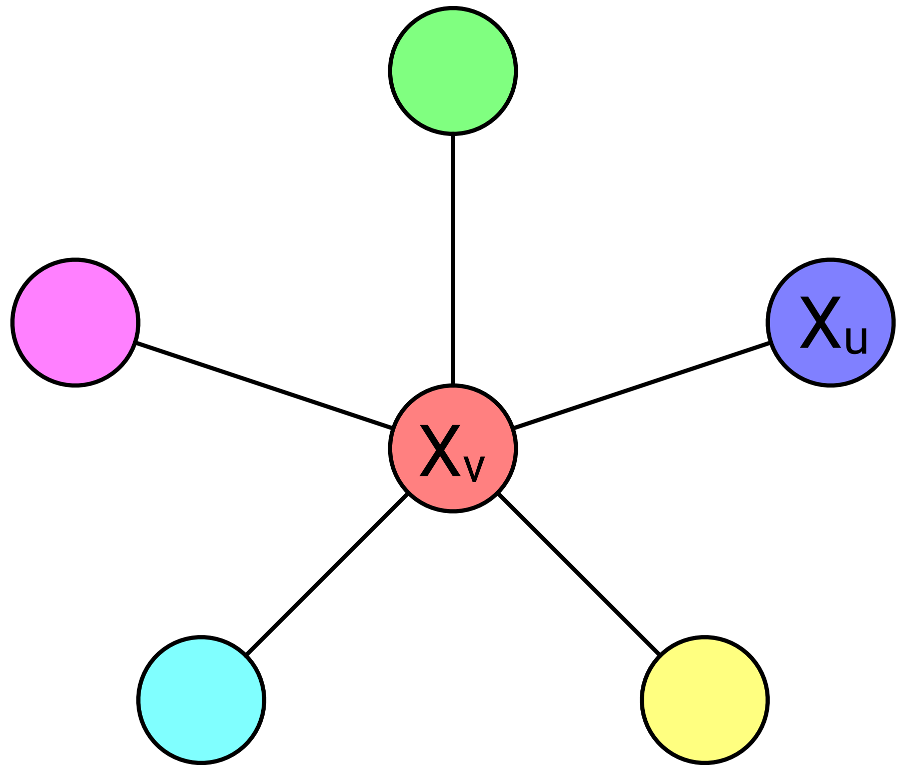


Random Colors

- **Lemma:** If each node $v \in V$ of a graph $G = (V,E)$ independently picks a **uniformly random** color X_v from $\{1, \ldots, \Delta+1\}$, for each node $v \in V$, the **probability** that $X_v \neq X_u$ for all neighbors u of v is **at least** $1/e$.
- P($X_v = X_u$)

  $= 1 / (\Delta+1)$
- P($X_v \neq X_u$)

  $= 1 - 1 / (\Delta+1)$
- P(X_v has a color **different** from **all neighbors**)



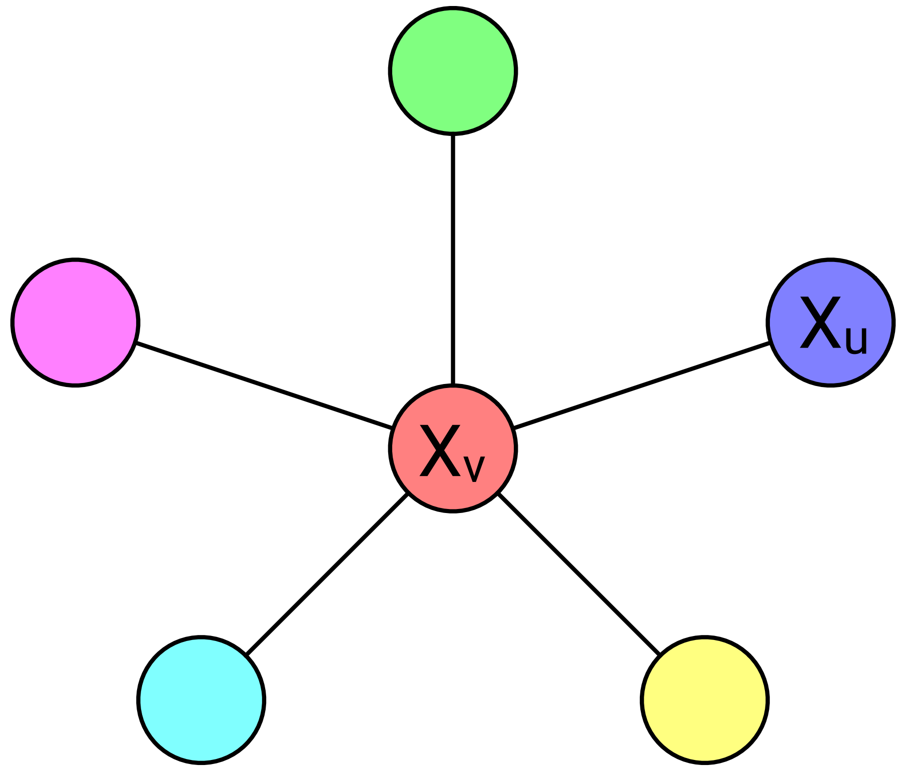
Random Colors

- **Lemma:** If each node $v \in V$ of a graph $G = (V,E)$ independently picks a **uniformly random** color X_v from $\{1, \ldots, \Delta+1\}$, for each node $v \in V$, the **probability** that $X_v \neq X_u$ for all neighbors u of v is **at least 1/e**.
- P($X_v = X_u$)

  $= 1 / (\Delta+1)$
- P($X_v \neq X_u$)

  $= 1 - 1 / (\Delta+1)$
- P(X_v has a color **different** from **all neighbors**)

  $\geq (1 - 1 / (\Delta+1))^{\Delta}$



Random Colors

- **Lemma:** If each node $v \in V$ of a graph $G = (V,E)$ independently picks a **uniformly random** color X_v from $\{1, ..., \Delta+1\}$, for each node $v \in V$, the **probability** that $X_v \neq X_u$ for all neighbors u of v is **at least 1/e**.

- P($X_v = X_u$)

  $= 1 / (\Delta+1)$

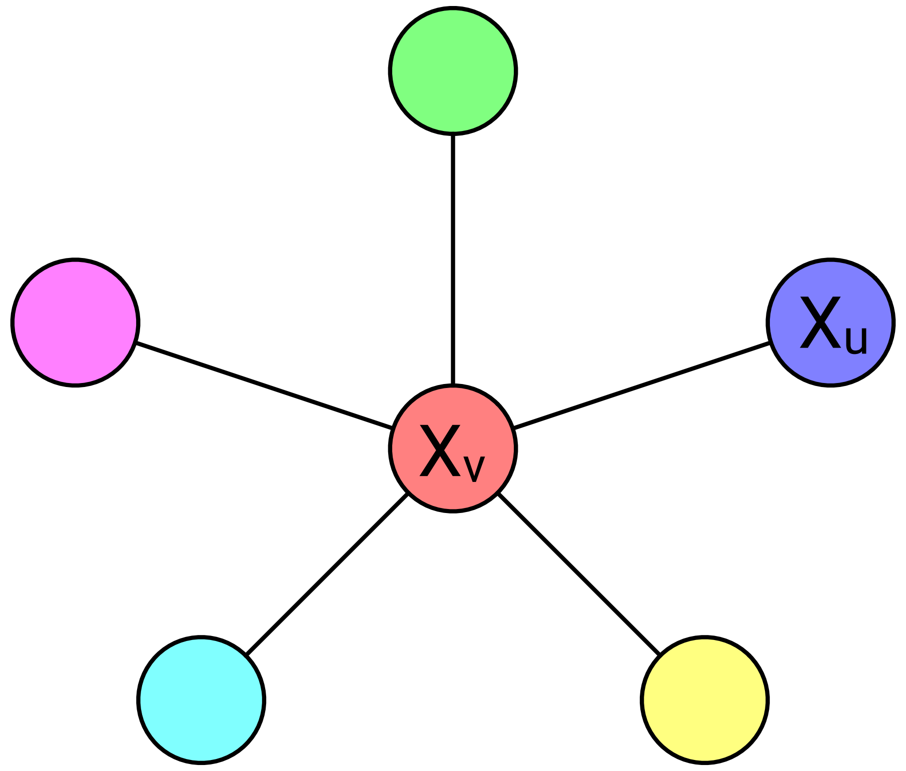
- P($X_v \neq X_u$)

  $= 1 - 1 / (\Delta+1)$

- P(X_v has a color **different** from **all neighbors**)

  $\geq (1 - 1 / (\Delta+1))^{\Delta}$

  $(1 - 1/(x+1))^x > 1/e$ for all $x > 0$



Random Colors

- **Lemma:** If each node $v \in V$ of a graph $G = (V,E)$ independently picks a **uniformly random** color X_v from $\{1, \ldots, \Delta+1\}$, for each node $v \in V$, the **probability** that $X_v \neq X_u$ for all neighbors u of v is **at least 1/e**.
- P($X_v = X_u$)

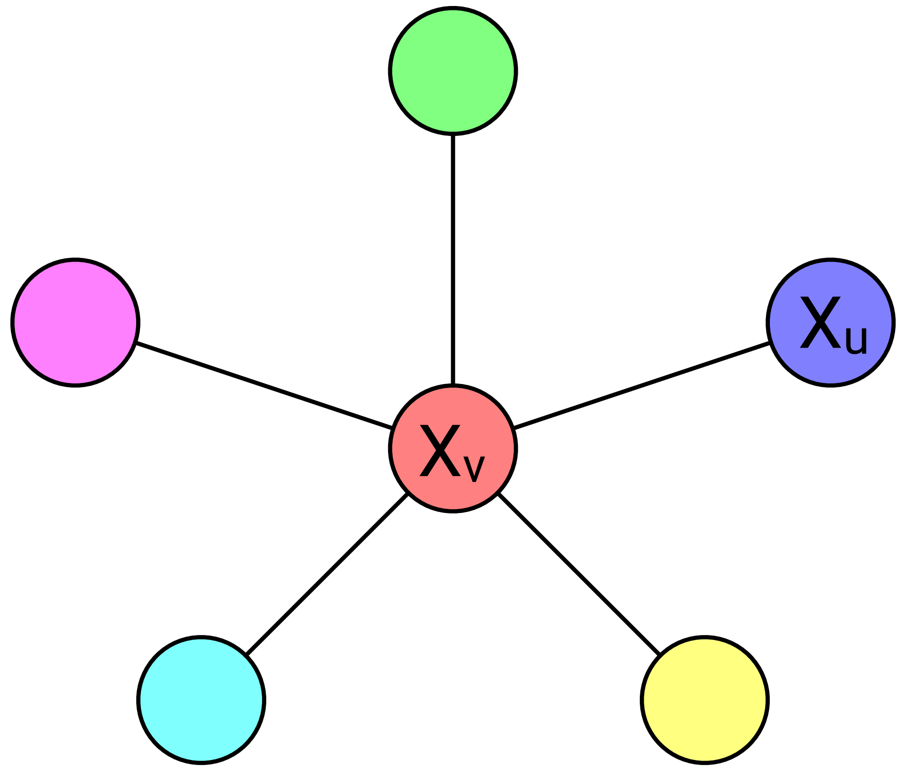
  $= 1 / (\Delta+1)$
- P($X_v \neq X_u$)

  $= 1 - 1 / (\Delta+1)$
- P(X_v has a color **different** from **all neighbors**)

  $\geq (1 - 1 / (\Delta+1))^{\Delta}$

  $> 1/e$

$(1 - 1/(x+1))^x > 1/e$ for all $x > 0$



Extending an Existing Coloring

Extending an Existing Coloring

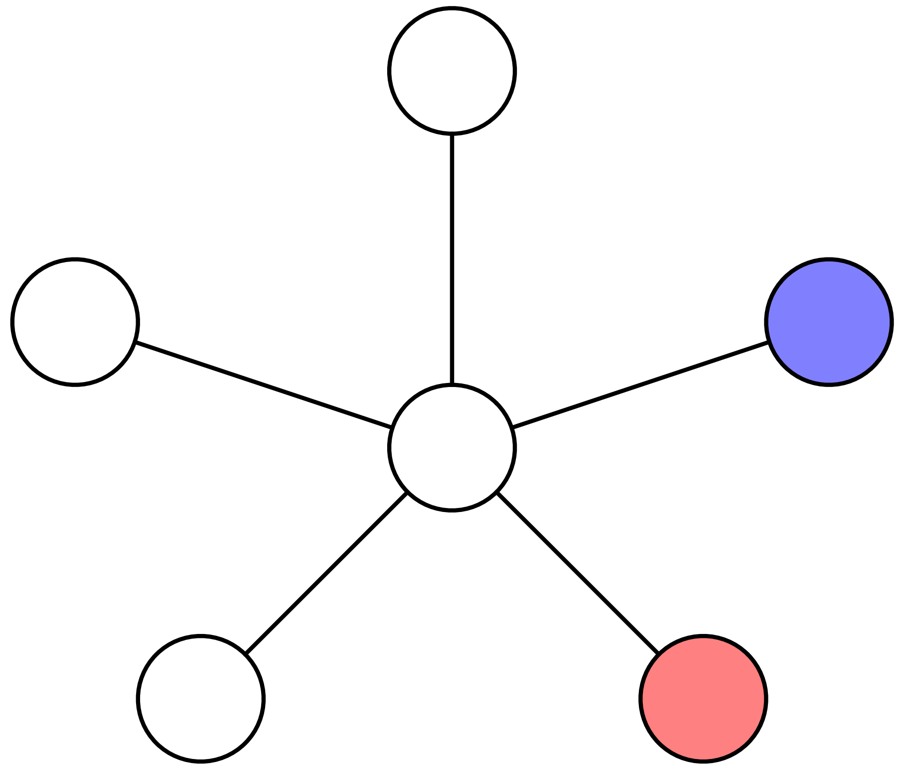
- After the first round, some nodes already picked a color (they were lucky with probability > 1/e)

Extending an Existing Coloring

- After the first round, some nodes already picked a color (they were lucky with probability > 1/e)
- In the second round, each node picks a random color

Extending an Existing Coloring

- After the first round, some nodes already picked a color (they were lucky with probability > 1/e)
- In the second round, each node picks a random color
  - It does not make sense to try colors already picked by the neighbors



Extending an Existing Coloring

- After the first round, some nodes already picked a color (they were lucky with probability > 1/e)

- In the second round, each node picks a random color

    - It does not make sense to try colors already picked by the neighbors

- **Is it still true** that for every possible color X_v, the **probability** that $X_v \neq X_u$ for all neighbors **u** of **v** is **at least 1/e?**

Extending an Existing Coloring

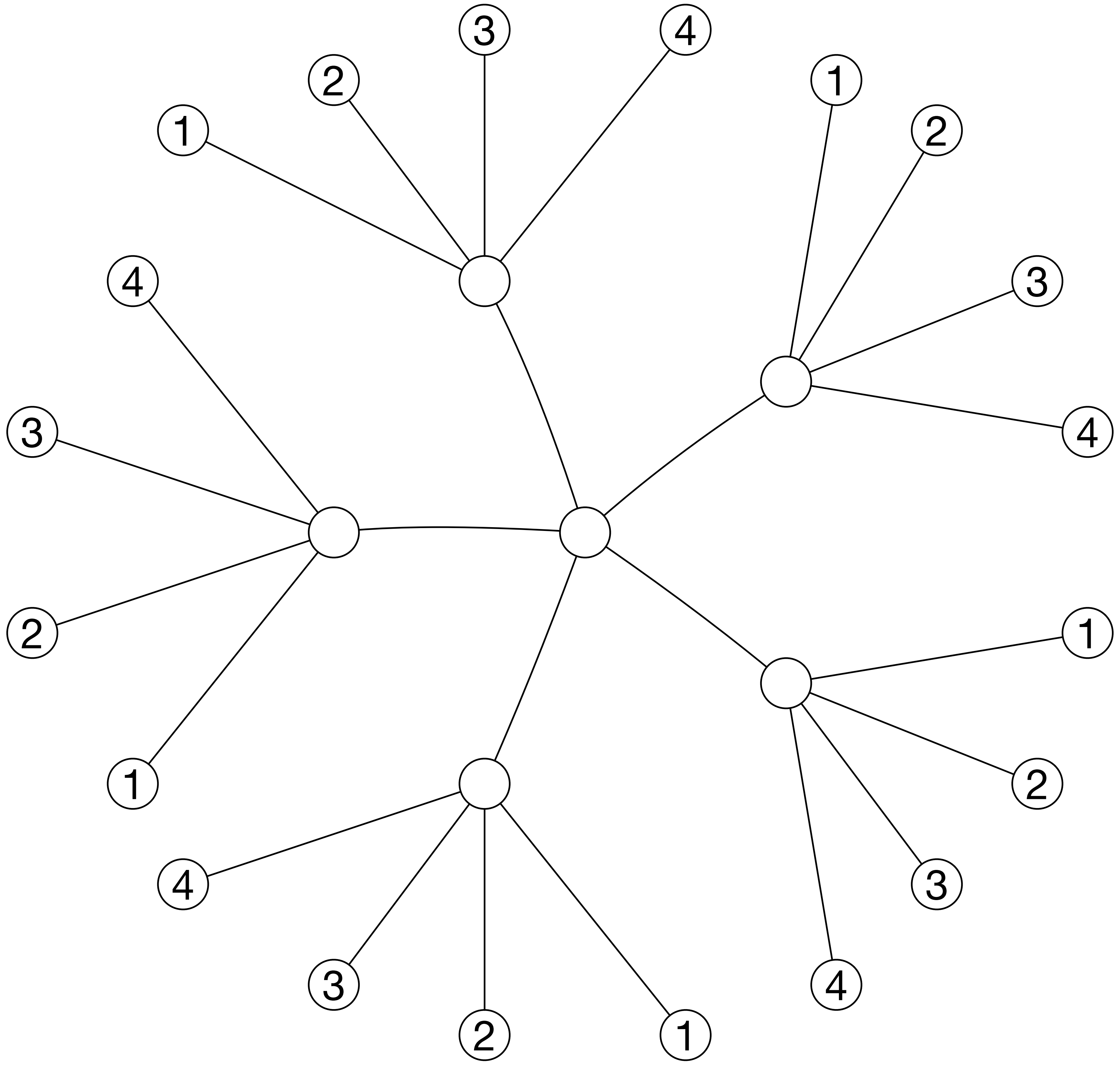
- After the first round, some nodes already picked a color (they were lucky with probability > 1/e)

- In the second round, each node picks a random color
  - It does not make sense to try colors already picked by the neighbors

- **Is it still true** that for every possible color X_v, the **probability** that $X_v \neq X_u$ for all neighbors **u** of **v** is **at least 1/e?**
  - **No!** Some colors are better than others.

Extending an Existing Coloring

- After the first round, some nodes already picked a color (they were lucky with probability > 1/e)
- In the second round, each node picks a random color
  - It does not make sense to try colors already picked by the neighbors
- **Is it still true** that for every possible color X_v, the **probability** that $X_v \neq X_u$ for all neighbors **u** of **v** is **at least 1/e?**
  - **No!** Some colors are better than others.

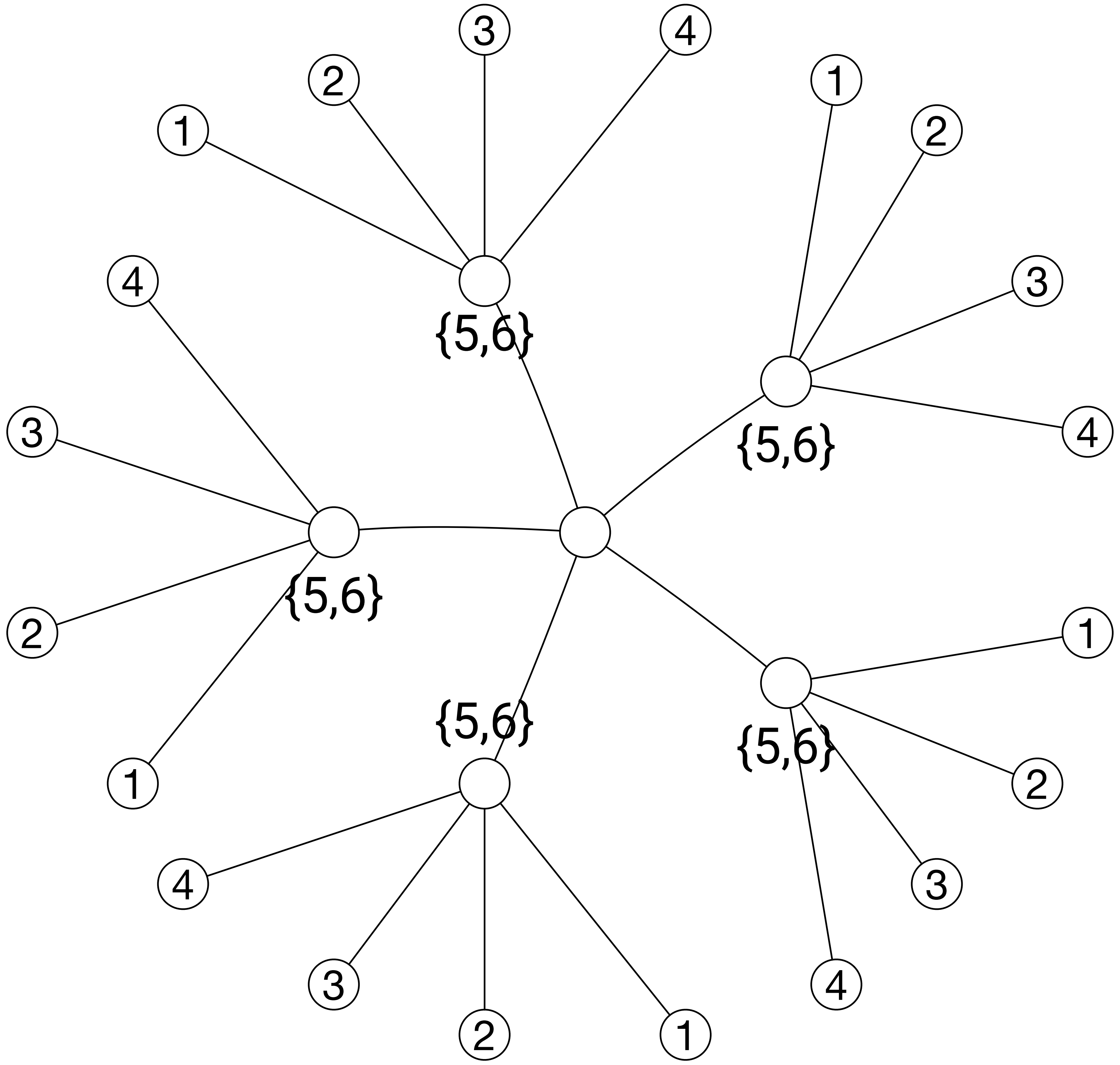


Extending an Existing Coloring

- After the first round, some nodes already picked a color (they were lucky with probability > 1/e)

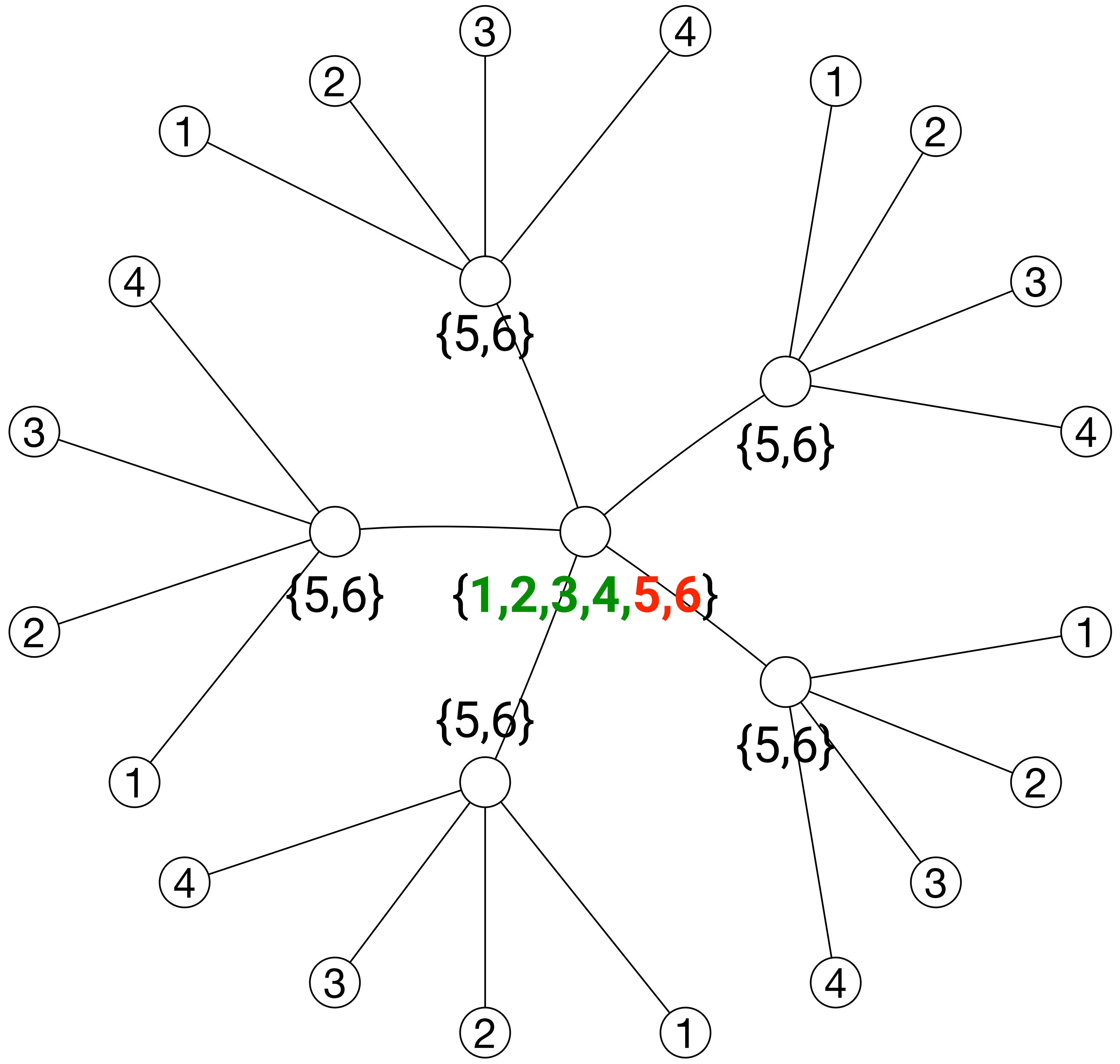
- In the second round, each node picks a random color
  - It does not make sense to try colors already picked by the neighbors

- **Is it still true** that for every possible color X_v, the **probability** that $X_v \neq X_u$ for all neighbors **u** of **v** is **at least 1/e?**
  - **No!** Some colors are better than others.



Extending an Existing Coloring

- After the first round, some nodes already picked a color (they were lucky with probability > 1/e)
- In the second round, each node picks a random color
  - It does not make sense to try colors already picked by the neighbors
- **Is it still true** that for every possible color X_v, the probability that $X_v \neq X_u$ for all neighbors u of v is at least 1/e?
  - **No!** Some colors are better than others.



Extending an Existing Coloring

Extending an Existing Coloring

- Let's make the problem harder and more precise

Extending an Existing Coloring

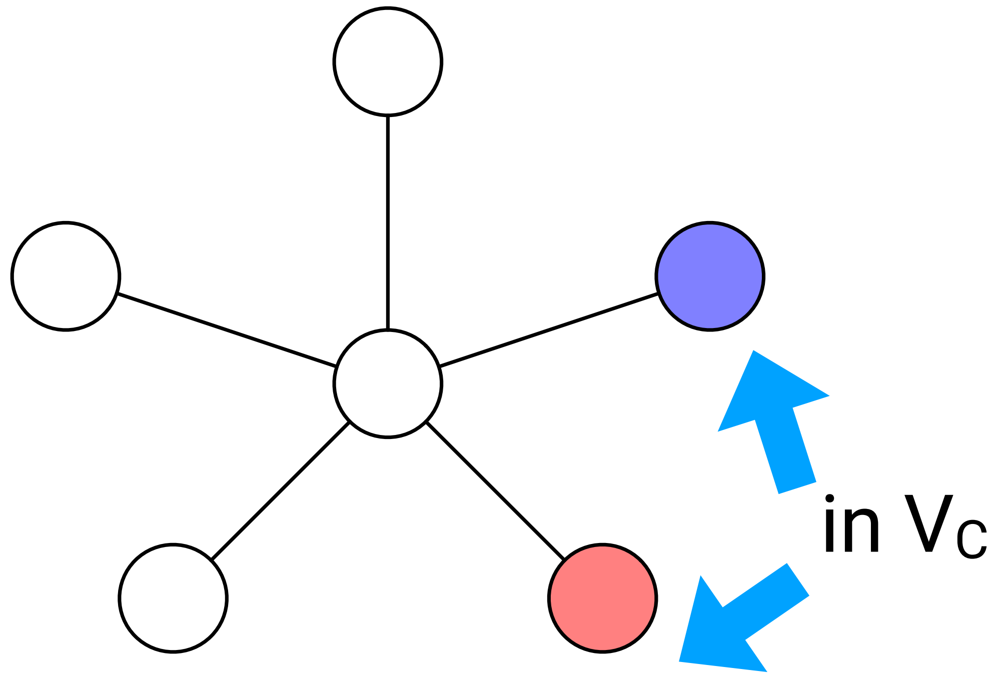
- Let's make the problem harder and more precise
- **Problem:**

Extending an Existing Coloring

- Let's make the problem harder and more precise
- **Problem:**
  - each node **$v \in V$** in **$G = (V, E)$** must get a color from **$\{1, ..., deg(v) + 1\}$**

Extending an Existing Coloring

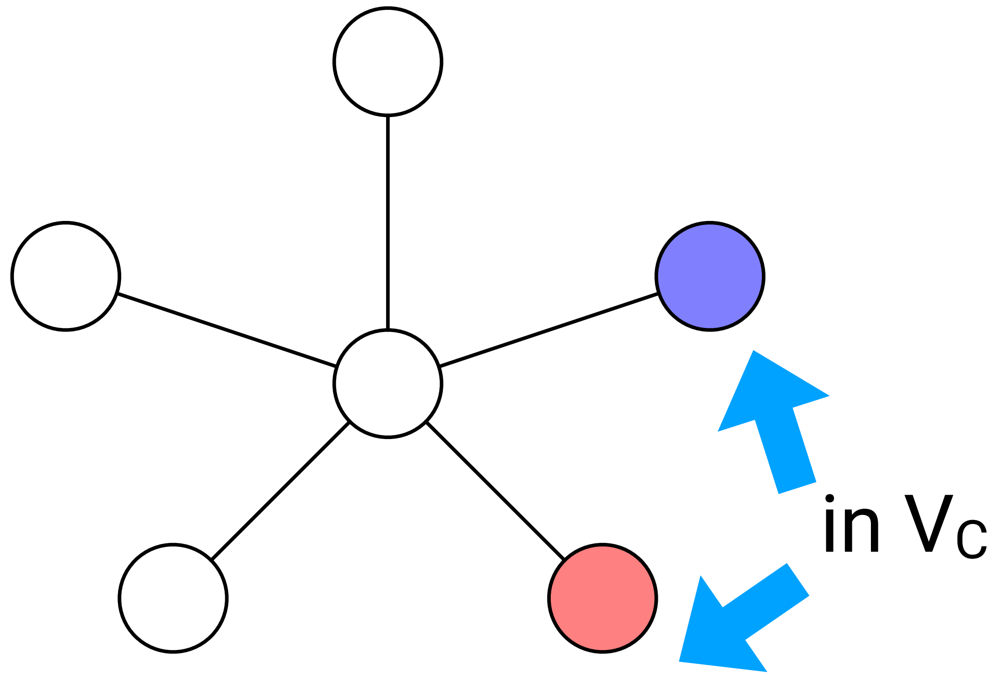
- Let's make the problem harder and more precise
- **Problem:**
  - each node $v \in V$ in $G = (V, E)$ must get a color from $\{1, \ldots, \deg(v) + 1\}$
- V_C: nodes $v \in V_C \subset V$ already have a color x_v such that $G[V_C]$ is properly colored



Extending an Existing Coloring

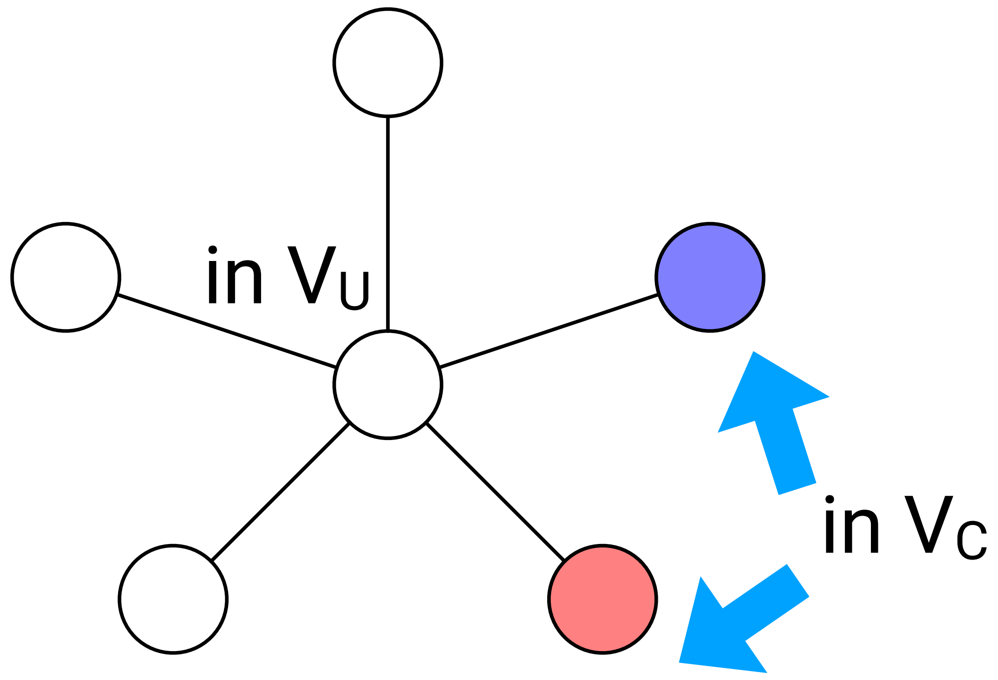
- Let's make the problem harder and more precise
- **Problem:**
  - each node $v \in V$ in $G = (V, E)$ must get a color from $\{1, ..., deg(v) + 1\}$
- V_C: nodes $v \in V_C \subset V$ already have a color x_v such that $G[V_C]$ is properly colored

Subgraph induced by nodes in V_C



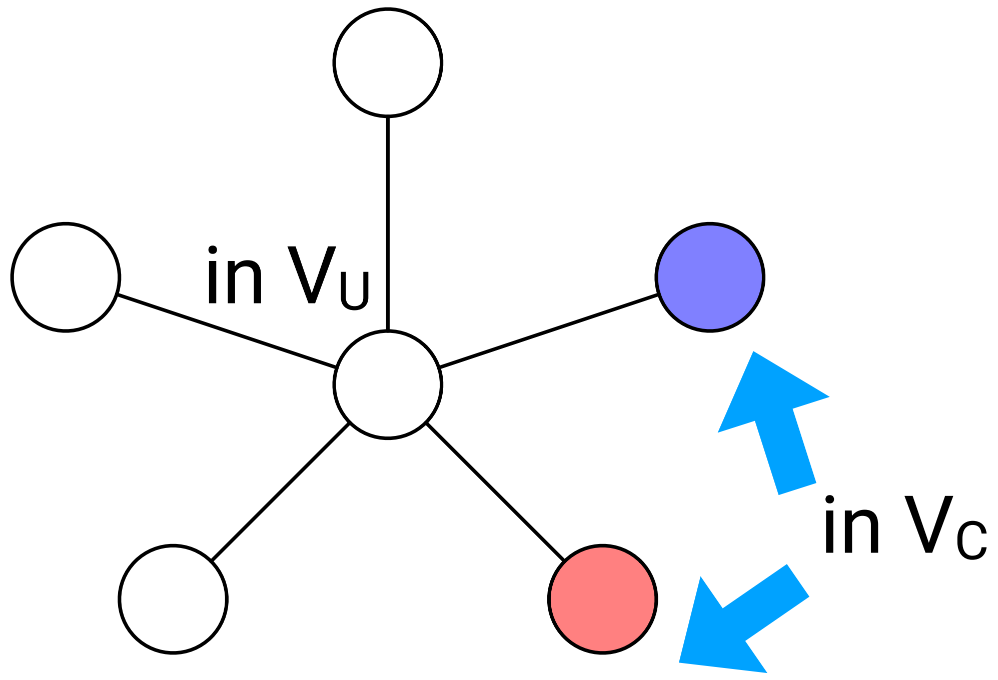
Extending an Existing Coloring

- Let's make the problem harder and more precise
- **Problem:**
  - each node $v \in V$ in $G = (V, E)$ must get a color from $\{1, ..., deg(v) + 1\}$
- V_C: nodes $v \in V_C \subset V$ already have a color x_v such that $G[V_C]$ is properly colored
- $V_U = V \setminus V_C$: set of uncolored nodes



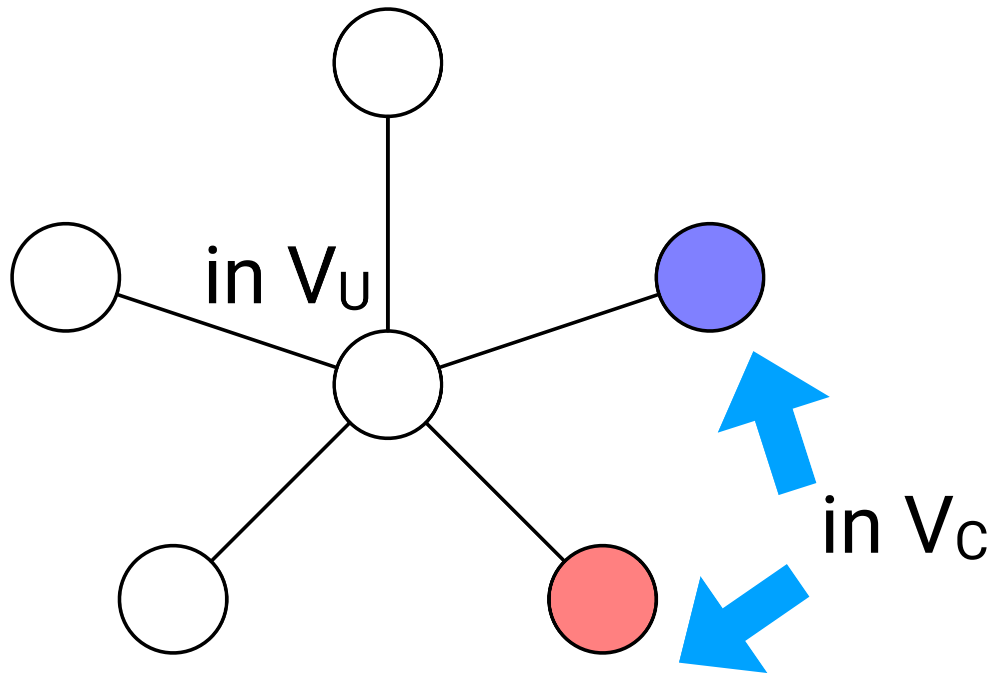
Extending an Existing Coloring

- Let's make the problem harder and more precise
- **Problem:**
  - each node $v \in V$ in $G = (V, E)$ must get a color from $\{1, ..., deg(v) + 1\}$
- V_C: nodes $v \in V_C \subset V$ already have a color x_v such that $G[V_C]$ is properly colored
- $V_U = V \setminus V_C$: set of uncolored nodes
- F_v: set of free colors of node v. $F_v = \{1, ..., deg(v)+1\} \setminus \bigcup_{u \in V_C \cap N(v)} \{ x_u \}$



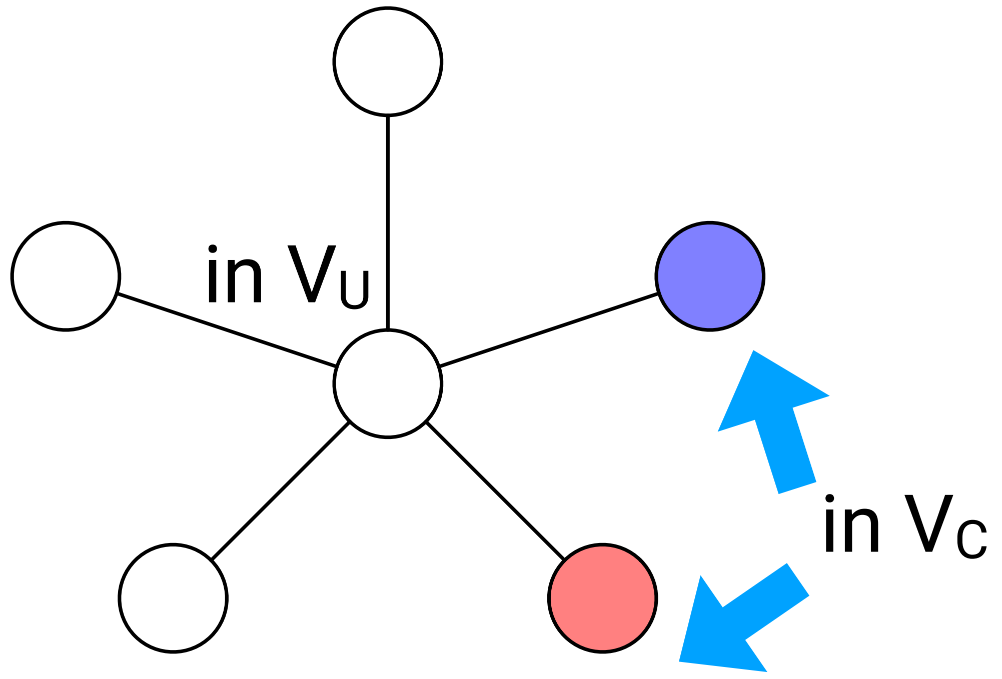
Extending an Existing Coloring

- Let's make the problem harder and more precise
- **Problem:**
  - each node $v \in V$ in $G = (V, E)$ must get a color from $\{1, ..., deg(v) + 1\}$
- V_C: nodes $v \in V_C \subset V$ already have a color x_v such that $G[V_C]$ is properly colored
- $V_U = V \setminus V_C$: set of uncolored nodes
- F_v: set of free colors of node v. $F_v = \underbrace{\{1, ..., deg(v)+1\}}_{\text{All colors}} \setminus \bigcup_{u \in V_C \cap N(v)} \{ x_u \}$



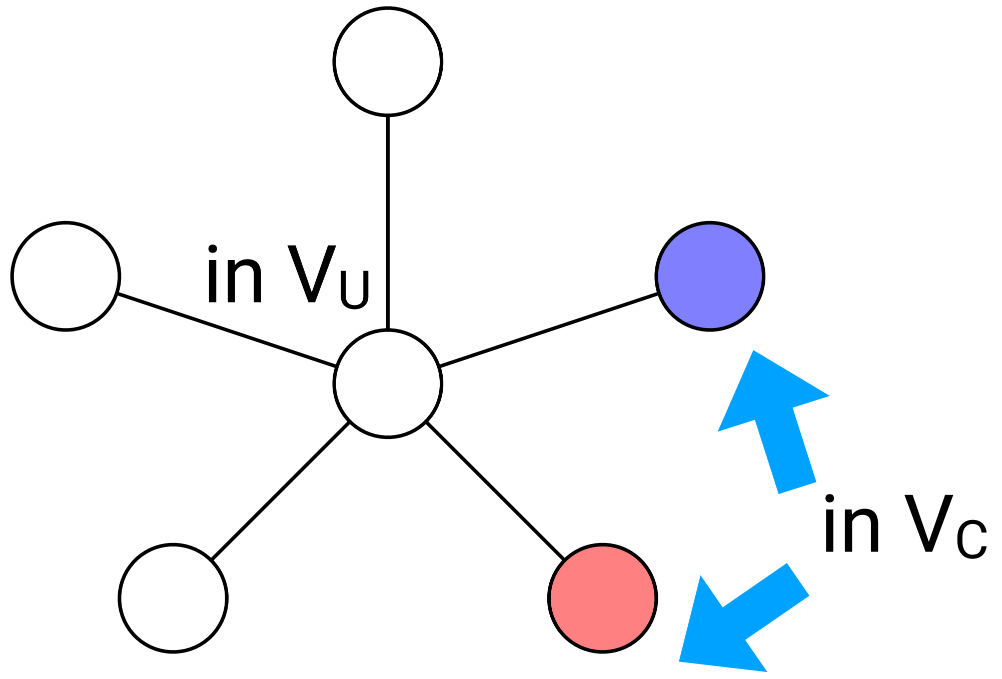
Extending an Existing Coloring

- Let's make the problem harder and more precise
- **Problem:**
  - each node $v \in V$ in $G = (V, E)$ must get a color from $\{1, ..., \deg(v) + 1\}$
- V_C: nodes $v \in V_C \subset V$ already have a color x_v such that $G[V_C]$ is properly colored
- $V_U = V \setminus V_C$: set of uncolored nodes
- F_v: set of free colors of node v. $F_v = \underbrace{\{1, ..., \deg(v)+1\}}_{\text{All colors}} \setminus \underbrace{\bigcup_{u \in V_C \cap N(v)}}_{\text{minus}} \{ x_u \}$



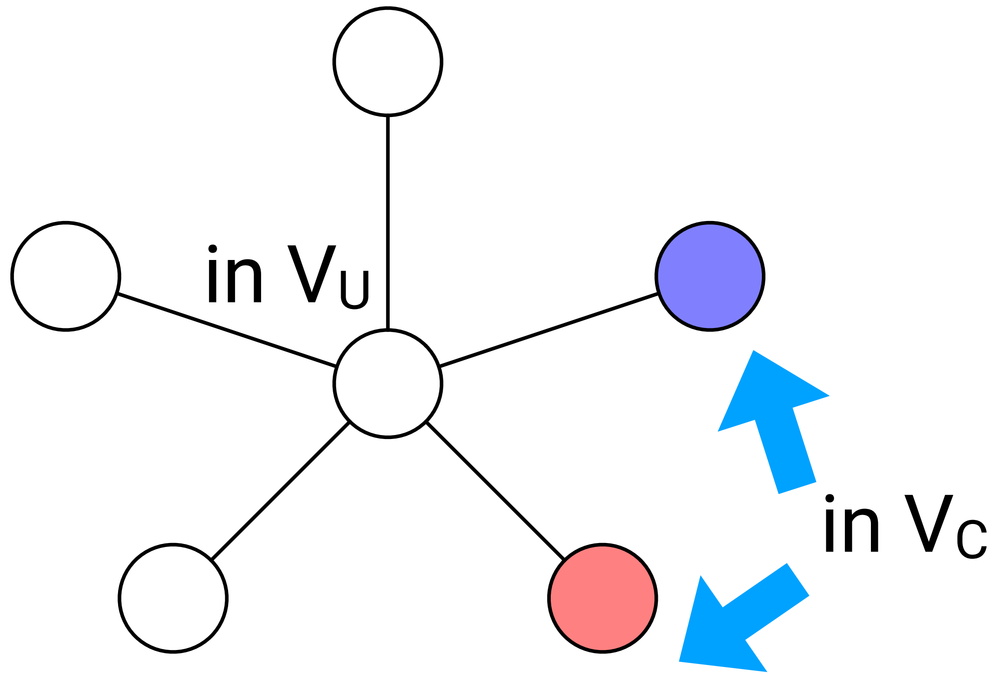
Extending an Existing Coloring

- Let's make the problem harder and more precise
- **Problem:**
  - each node $v \in V$ in $G = (V, E)$ must get a color from $\{1, ..., deg(v) + 1\}$
- V_C: nodes $v \in V_C \subset V$ already have a color x_v such that $G[V_C]$ is properly colored
- $V_U = V \setminus V_C$: set of uncolored nodes
- F_v: set of free colors of node v. $F_v = \underbrace{\{1, ..., deg(v)+1\}}_{\text{All colors}} \underbrace{\setminus}_{\text{minus}} \underbrace{\bigcup_{u \in V_C \cap N(v)} \{ x_u \}}_{\text{the colors used by the neighbors}}$



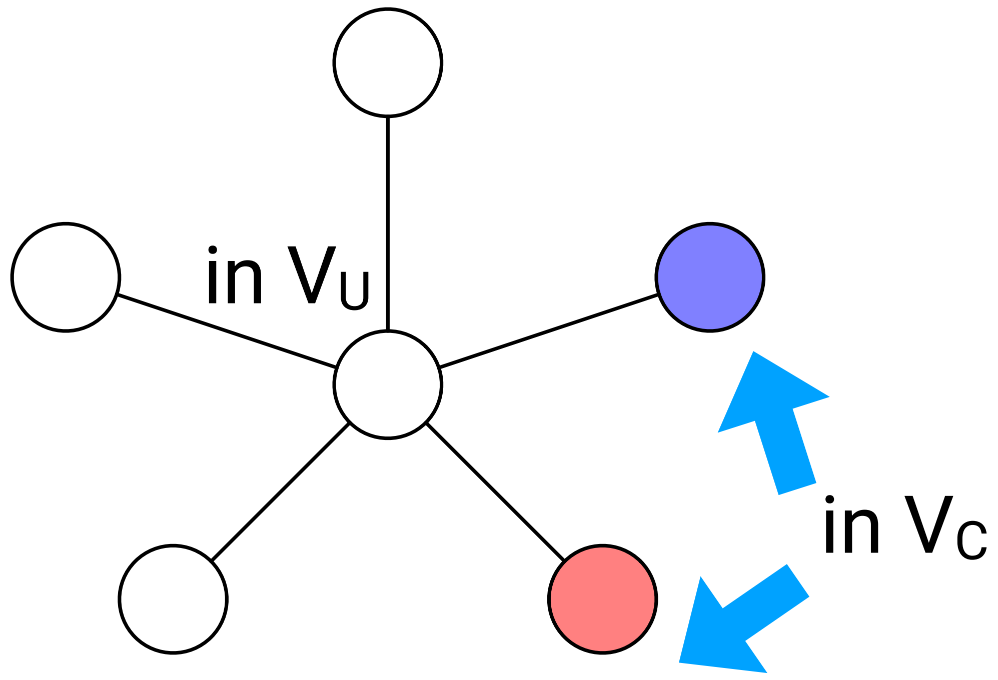
Extending an Existing Coloring

- Let's make the problem harder and more precise
- **Problem:**
  - each node $v \in V$ in $G = (V, E)$ must get a color from $\{1, ..., deg(v) + 1\}$
- V_C: nodes $v \in V_C \subset V$ already have a color x_v such that $G[V_C]$ is properly colored
- $V_U = V \setminus V_C$: set of uncolored nodes
- F_v: set of free colors of node v. $F_v = \{1, ..., deg(v)+1\} \setminus \bigcup_{u \in V_C \cap N(v)} \{ x_u \}$



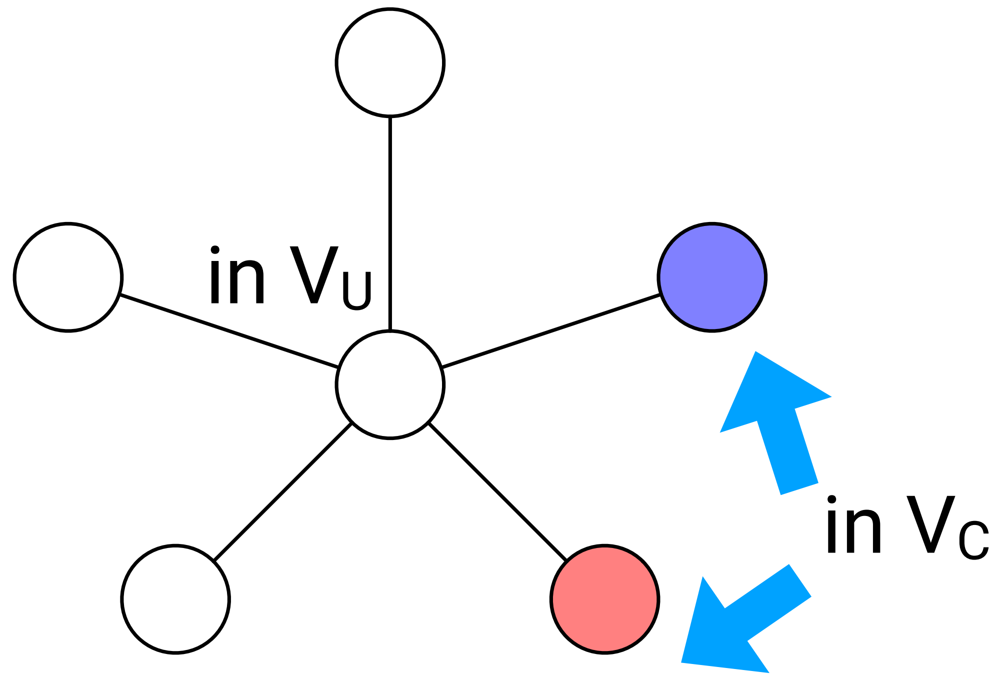
Extending an Existing Coloring

- Let's make the problem harder and more precise
- **Problem:**
  - each node $v \in V$ in $G = (V, E)$ must get a color from $\{1, \ldots, \deg(v) + 1\}$
- V_C: nodes $v \in V_C \subset V$ already have a color x_v such that $G[V_C]$ is properly colored
- $V_U = V \setminus V_C$: set of uncolored nodes
- F_v: set of free colors of node v. $F_v = \{1, \ldots, \deg(v)+1\} \setminus \bigcup_{u \in V_C \cap N(v)} \{ x_u \}$
- **Algorithm:**



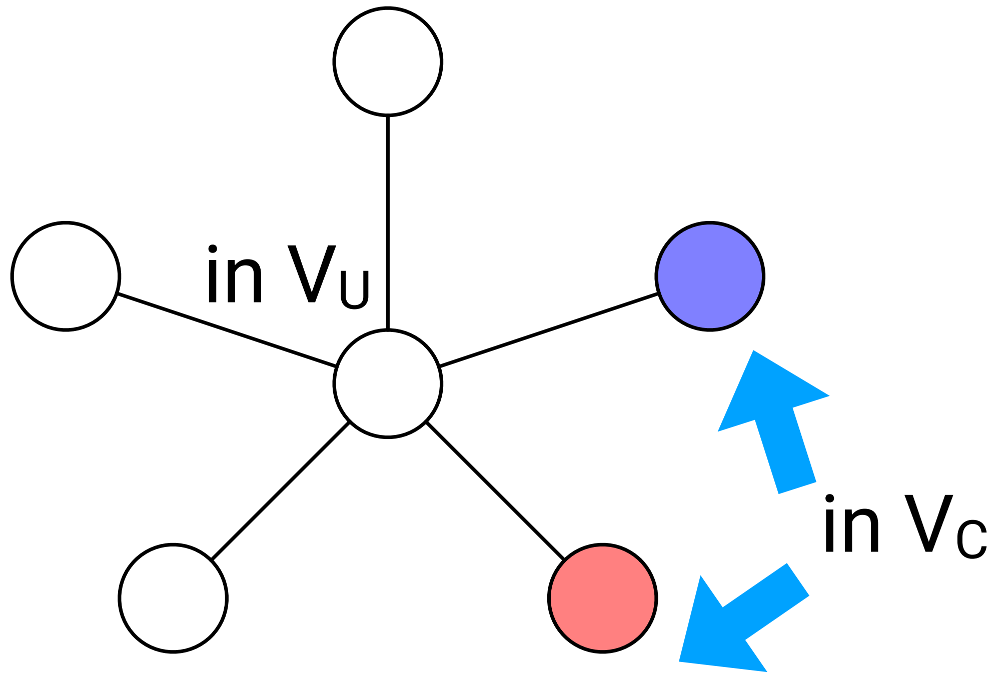
Extending an Existing Coloring

- Let's make the problem harder and more precise
- **Problem:**
  - each node $v \in V$ in $G = (V, E)$ must get a color from $\{1, ..., deg(v) + 1\}$
- V_C: nodes $v \in V_C \subset V$ already have a color x_v such that $G[V_C]$ is properly colored
- $V_U = V \setminus V_C$: set of uncolored nodes
- F_v: set of free colors of node v. $F_v = \{1, ..., deg(v)+1\} \setminus \bigcup_{u \in V_C \cap N(v)} \{ x_u \}$
- **Algorithm:**
  - Each node v picks a color uniformly at random from F_v



Extending an Existing Coloring

- Let's make the problem harder and more precise
- **Problem:**
  - each node $v \in V$ in $G = (V, E)$ must get a color from $\{1, ..., deg(v) + 1\}$
- V_C: nodes $v \in V_C \subset V$ already have a color x_v such that $G[V_C]$ is properly colored
- $V_U = V \setminus V_C$: set of uncolored nodes
- F_v: set of free colors of node v. $F_v = \{1, ..., deg(v)+1\} \setminus \bigcup_{u \in V_C \cap N(v)} \{ x_u \}$
- **Algorithm:**
  - Each node v picks a color uniformly at random from F_v
  - Each node $v \in V_U$ keeps the color if no neighbor in V_U picked the same color



Extending an Existing Coloring

Extending an Existing Coloring

- **$N_x(v)$**: **uncolored** neighbors **u** of **v** for which **$x \in F_u$**

Extending an Existing Coloring

- **$N_x(v)$**: **uncolored** neighbors **u** of **v** for which **$x \in F_u$** ← **Neighbors of v that could pick color x**

Extending an Existing Coloring

- **$N_x(v)$**: **uncolored** neighbors **u** of **v** for which **$x \in F_u$** ← **Neighbors of v that could pick color x**
- weight **$w_x(v)$** of a color **$x \in F_v$** for **$v \in V_U$**: $$w_x(v) = \sum_{u \in N_x(v)} 1 / |F_u|$$

Extending an Existing Coloring

- **$N_x(v)$**: **uncolored** neighbors **u** of **v** for which **$x \in F_u$**

Neighbors of v that could pick color x

- weight **$w_x(v)$** of a color **$x \in F_v$** for **$v \in V_U$**:

$$w_x(v) = \sum_{u \in N_x(v)} 1 / |F_u|$$

For each neighbor,
sum the probability that it takes color x

Extending an Existing Coloring

- **$N_x(v)$**: **uncolored** neighbors **u** of **v** for which **$x \in F_u$**

Neighbors of v that could pick color x

- weight **$w_x(v)$** of a color **$x \in F_v$** for **$v \in V_U$**:

$$w_x(v) = \sum_{u \in N_x(v)} 1 / |F_u|$$

For each neighbor,
sum the probability that it takes color x

- **Intuition:** $w_x(v)$ is **not** the probability that some neighbor of **v** picks color **x**,
but intuitively it **corresponds** to that

Extending an Existing Coloring

- **$N_x(v)$**: **uncolored** neighbors u of v for which $x \in F_u$

Neighbors of v that could pick color x

- weight **$w_x(v)$** of a color $x \in F_v$ for $v \in V_U$: $w_x(v) = \sum_{u \in N_x(v)} 1 / |F_u|$

For each neighbor,
sum the probability that it takes color x

- **Intuition:** $w_x(v)$ is **not** the probability that some neighbor of v picks color x, but intuitively it **corresponds** to that

- **Lemma:** $\sum_{x \in F_v} w_x(v) \leq |N(v) \cap V_U|$

Extending an Existing Coloring

- $N_x(v)$: **uncolored** neighbors u of v for which $x \in F_u$

Neighbors of v that could pick color x

- weight $w_x(v)$ of a color $x \in F_v$ for $v \in V_U$: $$w_x(v) = \sum_{u \in N_x(v)} 1 / |F_u|$$

For each neighbor,
sum the probability that it takes color x

- **Intuition:** $w_x(v)$ is **not** the probability that some neighbor of v picks color x, but intuitively it **corresponds** to that

- **Lemma:** $$\sum_{x \in F_v} w_x(v) \le |N(v) \cap V_U|$$

The sum of the weights is at most
the number of uncolored neighbors

Extending an Existing Coloring

- **$N_x(v)$**: **uncolored** neighbors **u** of **v** for which **$x \in F_u$** ← **Neighbors of v that could pick color x**
- weight **$w_x(v)$** of a color **$x \in F_v$** for **$v \in V_U$**: $$w_x(v) = \sum_{u \in N_x(v)} 1 / |F_u|$$ **For each neighbor, sum the probability that it takes color x**
- **Intuition:** $w_x(v)$ is **not** the probability that some neighbor of **v** picks color **x**, but intuitively it **corresponds** to that
- **Lemma:** $$\sum_{x \in F_v} w_x(v) \leq |N(v) \cap V_U| \qquad \sum_{x \in F_v} w_x(v)$$

The sum of the weights is at most the number of uncolored neighbors

Extending an Existing Coloring

- **$N_x(v)$**: **uncolored** neighbors u of v for which $x \in F_u$

Neighbors of v that could pick color x

- weight **$w_x(v)$** of a color $x \in F_v$ for $v \in V_U$:

$$w_x(v) = \sum_{u \in N_x(v)} 1 / |F_u|$$

For each neighbor,
sum the probability that it takes color x

- **Intuition:** $w_x(v)$ is **not** the probability that some neighbor of v picks color x, but intuitively it **corresponds** to that

- **Lemma:** $\sum_{x \in F_v} w_x(v) \leq |N(v) \cap V_U|$

$$\sum_{x \in F_v} w_x(v) = \sum_{x \in F_v} \sum_{u \in N_x(v)} 1 / |F_u|$$

The sum of the weights is at most
the number of uncolored neighbors

Extending an Existing Coloring

- $N_x(v)$: **uncolored** neighbors u of v for which $x \in F_u$

Neighbors of v that could pick color x

- weight $w_x(v)$ of a color $x \in F_v$ for $v \in V_U$: $$w_x(v) = \sum_{u \in N_x(v)} 1 / |F_u|$$

For each neighbor,
sum the probability that it takes color x

- **Intuition:** $w_x(v)$ is **not** the probability that some neighbor of v picks color x,
  but intuitively it **corresponds** to that

- **Lemma:** $$\sum_{x \in F_v} w_x(v) \leq |N(v) \cap V_U|$$

The sum of the weights is at most
the number of uncolored neighbors

$$\sum_{x \in F_v} w_x(v) = \sum_{x \in F_v} \sum_{u \in N_x(v)} 1 / |F_u|$$

$$= \sum_{u \in N(v) \cap V_U} \sum_{x \in F_v \cap F_u} 1 / |F_u|$$

Extending an Existing Coloring

- **$N_x(v)$**: **uncolored** neighbors u of v for which $x \in F_u$ ← **Neighbors of v that could pick color x**
- weight **$w_x(v)$** of a color $x \in F_v$ for $v \in V_U$: $$w_x(v) = \sum_{u \in N_x(v)} 1 / |F_u|$$ **For each neighbor, sum the probability that it takes color x**
- **Intuition:** $w_x(v)$ is **not** the probability that some neighbor of v picks color x, but intuitively it **corresponds** to that
- **Lemma:** $$\sum_{x \in F_v} w_x(v) \le |N(v) \cap V_U|$$ **The sum of the weights is at most the number of uncolored neighbors**

$$\sum_{x \in F_v} w_x(v) = \sum_{x \in F_v} \sum_{u \in N_x(v)} 1 / |F_u|$$

$$= \sum_{u \in N(v) \cap V_U} \sum_{x \in F_v \cap F_u} 1 / |F_u|$$

Uncolored neighbors

Extending an Existing Coloring

- **$N_x(v)$**: uncolored neighbors u of v for which $x \in F_u$

Neighbors of v that could pick color x

- weight **$w_x(v)$** of a color $x \in F_v$ for $v \in V_U$: $\quad w_x(v) = \sum_{u \in N_x(v)} 1 / |F_u|$

**For each neighbor,
sum the probability that it takes color x**

- **Intuition:** $w_x(v)$ is **not** the probability that some neighbor of v picks color x, but intuitively it **corresponds** to that

- **Lemma:** $\sum_{x \in F_v} w_x(v) \leq |N(v) \cap V_U|$

**The sum of the weights is at most
the number of uncolored neighbors**

$$\sum_{x \in F_v} w_x(v) = \sum_{x \in F_v} \sum_{u \in N_x(v)} 1 / |F_u|$$

$$= \sum_{u \in N(v) \cap V_U} \sum_{x \in F_v \cap F_u} 1 / |F_u|$$

Uncolored neighbors **Common colors**

Extending an Existing Coloring

- **$N_x(v)$**: **uncolored** neighbors u of v for which $x \in F_u$

Neighbors of v that could pick color x

- weight **$w_x(v)$** of a color $x \in F_v$ for $v \in V_U$: $\quad w_x(v) = \sum_{u \in N_x(v)} 1 / |F_u|$

**For each neighbor,
sum the probability that it takes color x**

- **Intuition:** $w_x(v)$ is **not** the probability that some neighbor of v picks color x, but intuitively it **corresponds** to that

- **Lemma:** $\sum_{x \in F_v} w_x(v) \leq |N(v) \cap V_U|$

**The sum of the weights is at most
the number of uncolored neighbors**

$$\sum_{x \in F_v} w_x(v) = \sum_{x \in F_v} \sum_{u \in N_x(v)} 1 / |F_u|$$

$$= \sum_{u \in N(v) \cap V_U} \sum_{x \in F_v \cap F_u} 1 / |F_u| = \sum_{u \in N(v) \cap V_U} |F_v \cap F_u| / |F_u|$$

Uncolored neighbors **Common colors**

Extending an Existing Coloring

- **$N_x(v)$**: **uncolored** neighbors u of v for which $x \in F_u$

Neighbors of v that could pick color x

- weight **$w_x(v)$** of a color $x \in F_v$ for $v \in V_U$: $\quad w_x(v) = \sum_{u \in N_x(v)} 1 / |F_u|$

For each neighbor,
sum the probability that it takes color x

- **Intuition:** $w_x(v)$ is **not** the probability that some neighbor of v picks color x, but intuitively it **corresponds** to that

- **Lemma:** $\sum_{x \in F_v} w_x(v) \le |N(v) \cap V_U|$

The sum of the weights is at most
the number of uncolored neighbors

$$\sum_{x \in F_v} w_x(v) = \sum_{x \in F_v} \sum_{u \in N_x(v)} 1 / |F_u|$$

$$= \underbrace{\sum_{u \in N(v) \cap V_U}}_{\text{Uncolored neighbors}} \underbrace{\sum_{x \in F_v \cap F_u}}_{\text{Common colors}} 1 / |F_u| = \sum_{u \in N(v) \cap V_U} |F_v \cap F_u| / |F_u|$$

$$\le \sum_{u \in N(v) \cap V_U} 1$$

Extending an Existing Coloring

- **$N_x(v)$**: **uncolored** neighbors **u** of **v** for which $x \in F_u$

Neighbors of v that could pick color x

- weight **$w_x(v)$** of a color $x \in F_v$ for $v \in V_U$: $w_x(v) = \sum_{u \in N_x(v)} 1 / |F_u|$

For each neighbor, sum the probability that it takes color x

- **Intuition:** $w_x(v)$ is **not** the probability that some neighbor of **v** picks color **x**, but intuitively it **corresponds** to that

- **Lemma:** $\sum_{x \in F_v} w_x(v) \leq |N(v) \cap V_U|$

The sum of the weights is at most the number of uncolored neighbors

$$\sum_{x \in F_v} w_x(v) = \sum_{x \in F_v} \sum_{u \in N_x(v)} 1 / |F_u|$$

$$= \sum_{u \in N(v) \cap V_U} \sum_{x \in F_v \cap F_u} 1 / |F_u| = \sum_{u \in N(v) \cap V_U} |F_v \cap F_u| / |F_u|$$

$$\leq \sum_{u \in N(v) \cap V_U} 1 = |N(v) \cap V_U|$$

Uncolored neighbors **Common colors**

Extending an Existing Coloring

Extending an Existing Coloring

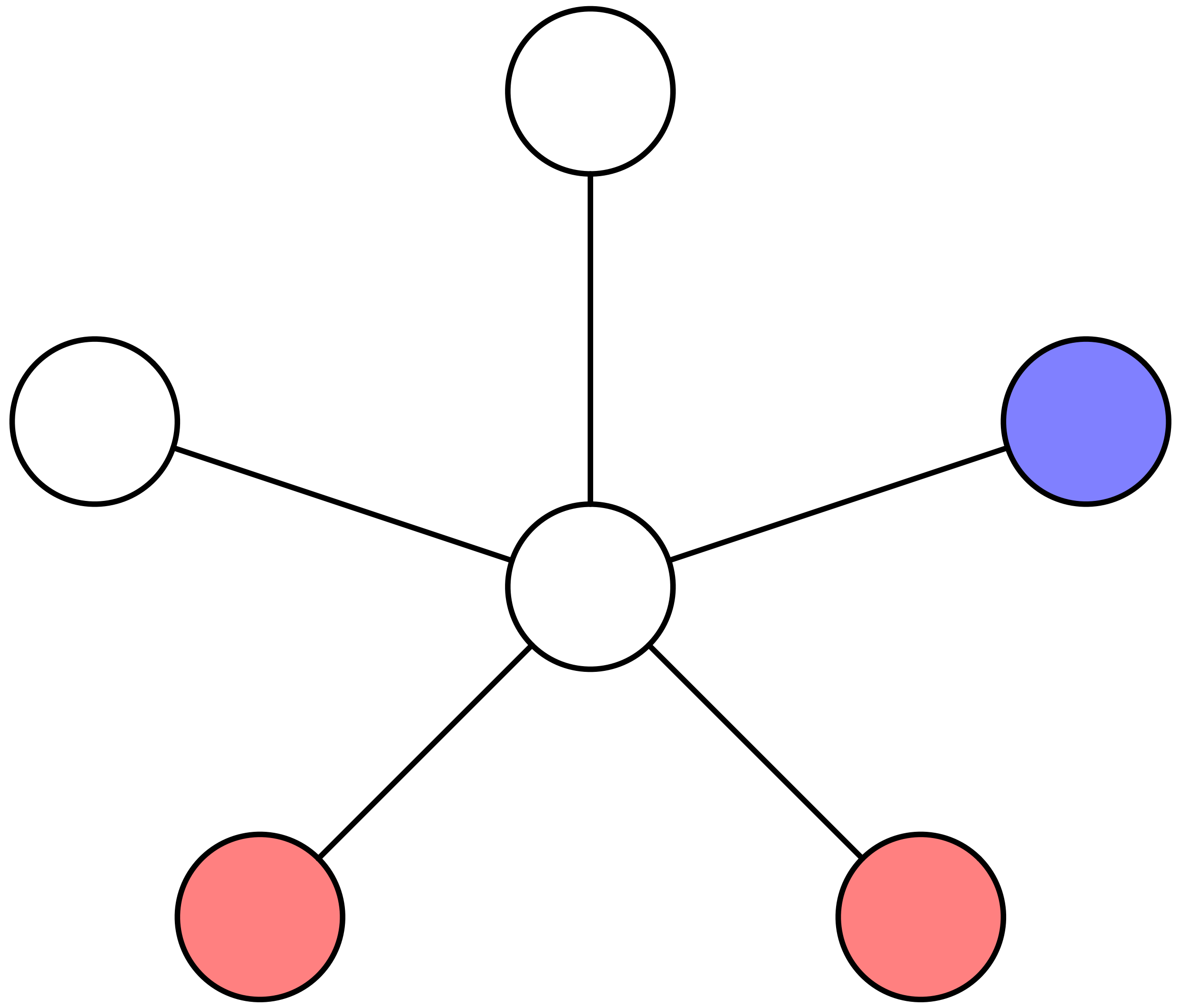
- **Lemma:** $\sum_{x \in F_v} w_x(v) \leq |N(v) \cap V_U|$

Extending an Existing Coloring

- **Lemma:** $\sum_{x \in F_v} w_x(v) \leq |N(v) \cap V_U|$
- **Corollary:** $\sum_{x \in F_v} w_x(v) \leq |N(v) \cap V_U| \leq |F_v| - 1$

Extending an Existing Coloring

- **Lemma:** $\sum_{x \in F_v} w_x(v) \leq |N(v) \cap V_U|$
- **Corollary:** $\sum_{x \in F_v} w_x(v) \leq |N(v) \cap V_U| \leq |F_v| - 1$



Extending an Existing Coloring

Extending an Existing Coloring

- **Lemma:** If node $v \in V_U$ picks the random color $x \in F_v$, the probability that v can keep its colors is at least $4^{-w_x(v)}$

Extending an Existing Coloring

- **Lemma:** If node $v \in V_U$ picks the random color $x \in F_v$, the probability that v can keep its colors is at least $4^{-w_x(v)}$

- **P(** v can keep its color x **)**

Extending an Existing Coloring

- **Lemma:** If node $v \in V_U$ picks the random color $x \in F_v$, the probability that v can keep its colors is at least $4^{-w_x(v)}$

- **P(** v can keep its color x **)**

$$= \prod_{u \in N_x(v)} (1 - 1 / |F_u|)$$

Extending an Existing Coloring

- **Lemma:** If node $v \in V_U$ picks the random color $x \in F_v$, the probability that v can keep its colors is at least $4^{-w_x(v)}$

- **P(** v can keep its color x **)**

$$= \prod_{u \in N_x(v)} (1 - 1 / |F_u|)$$

Neighbors that could also pick x

Extending an Existing Coloring

- **Lemma:** If node $v \in V_U$ picks the random color $x \in F_v$, the probability that v can keep its colors is at least $4^{-w_x(v)}$

- **P(** v can keep its color x **)**

$$= \prod_{u \in N_x(v)} (1 - 1 / |F_u|)$$

Probability that u does **not** pick x

Neighbors that could also pick x

Extending an Existing Coloring

- **Lemma:** If node $v \in V_U$ picks the random color $x \in F_v$, the probability that v can keep its colors is at least $4^{-w_x(v)}$

- **P(** v can keep its color x **)**

$$= \prod_{u \in N_x(v)} (1 - 1 / |F_u|)$$

Probability that u does **not** pick x

Neighbors that could also pick x

$(1 - x) \geq 4^{-x}$ for all $x \in [0, 1/2]$

Extending an Existing Coloring

- **Lemma:** If node $v \in V_U$ picks the random color $x \in F_v$, the probability that v can keep its colors is at least $4^{-w_x(v)}$

- **P(** v can keep its color x **)**

$$= \prod_{u \in N_x(v)} (1 - 1 / |F_u|)$$

Probability that u does **not** pick x

Neighbors that could also pick x

$(1 - x) \geq 4^{-x}$ for all $x \in [0, 1/2]$

$|F_u| \geq 2$, otherwise u has no uncolored neighbors

Extending an Existing Coloring

- **Lemma:** If node $v \in V_U$ picks the random color $x \in F_v$, the probability that v can keep its colors is at least $4^{-w_x(v)}$

- **P(** v can keep its color x **)**

$$= \prod_{u \in N_x(v)} (1 - 1 / |F_u|) \geq \prod_{u \in N_x(v)} 4^{-1 / |F_u|}$$

Probability that u does **not** pick x

Neighbors that could also pick x

$(1 - x) \geq 4^{-x}$ for all $x \in [0, 1/2]$

$|F_u| \geq 2$, otherwise u has no uncolored neighbors

Extending an Existing Coloring

- **Lemma:** If node $v \in V_U$ picks the random color $x \in F_v$, the probability that v can keep its colors is at least $4^{-w_x(v)}$
- **P(** v can keep its color x **)**

$$= \prod_{u \in N_x(v)} (1 - 1 / |F_u|) \ge \prod_{u \in N_x(v)} 4^{-1 / |F_u|} = 4^{-\sum_{u \in N_x(v)} 1 / |F_u|}$$

Probability that u does **not** pick x

Neighbors that could also pick x

$(1 - x) \ge 4^{-x}$ for all $x \in [0, 1/2]$

$|F_u| \ge 2$, otherwise u has no uncolored neighbors

Extending an Existing Coloring

- **Lemma:** If node $v \in V_U$ picks the random color $x \in F_v$, the probability that v can keep its colors is at least $4^{-w_x(v)}$

- **P(** v can keep its color x **)**

$$= \prod_{u \in N_x(v)} (1 - 1 / |F_u|) \geq \prod_{u \in N_x(v)} 4^{-1 / |F_u|} = 4^{-\sum_{u \in N_x(v)} 1 / |F_u|} = 4^{-w_x(v)}$$

Probability that u does **not** pick x

Neighbors that could also pick x

$(1 - x) \geq 4^{-x}$ for all $x \in [0, 1/2]$

$|F_u| \geq 2$, otherwise u has no uncolored neighbors

Extending an Existing Coloring

Extending an Existing Coloring

- **Theorem:** The probability that a node $v \in V_U$ can keep its random color is at least $1/4$

Extending an Existing Coloring

- **Theorem:** The probability that a node $v \in V_U$ can keep its random color is at least $1/4$
- P(v can keep its color X_v)

Extending an Existing Coloring

- **Theorem:** The probability that a node $v \in V_U$ can keep its random color is at least $1/4$
- $P(\, v$ can keep its color $X_v\,)$

$$= \sum_{x \in F_v} 1 / |F_v| \cdot P(\, v \text{ can keep color } x \mid X_v = x\,)$$

Extending an Existing Coloring

- **Theorem:** The probability that a node $v \in V_U$ can keep its random color is at least $1/4$

- P(v can keep its color X_v)

$$= \sum_{x \in F_v} 1 / |F_v| \cdot P(\, v \text{ can keep color } x \mid X_v = x \,)$$

$$\geq \sum_{x \in F_v} 1 / |F_v| \cdot 4^{-w_x(v)}$$

Extending an Existing Coloring

- **Theorem:** The probability that a node $v \in V_U$ can keep its random color is at least $1/4$
- P(v can keep its color X_v)

$$= \sum_{x \in F_v} 1 / |F_v| \cdot P(\text{ } v \text{ can keep color } x \mid X_v = x \text{ })$$

$$\geq \sum_{x \in F_v} 1 / |F_v| \cdot 4^{-w_x(v)}$$

$$\geq 4^{-1 / |F_v| \cdot \sum_{x \in F_v} w_x(v)}$$

Extending an Existing Coloring

- **Theorem:** The probability that a node $v \in V_U$ can keep its random color is at least $1/4$
- $P(v \text{ can keep its color } X_v)$

$$= \sum_{x \in F_v} 1 / |F_v| \cdot P(v \text{ can keep color } x \mid X_v = x)$$

$$\geq \sum_{x \in F_v} 1 / |F_v| \cdot 4^{- w_x(v)}$$

Let $f(w_x) = 4^{-w_x(v)}$. This is $\text{average}(f(w_x))$

$$\geq 4^{- 1 / |F_v| \cdot \sum_{x \in F_v} w_x(v)}$$

Extending an Existing Coloring

- **Theorem:** The probability that a node $v \in V_U$ can keep its random color is at least 1/4
- $P(\ v \text{ can keep its color } X_v\ )$

$$= \sum_{x \in F_v} 1 / |F_v| \cdot P(\ v \text{ can keep color } x \mid X_v = x\ )$$

$$\geq \sum_{x \in F_v} 1 / |F_v| \cdot 4^{-w_x(v)}$$

Let $f(w_x) = 4^{-w_x(v)}$. This is average($f(w_x)$)

$$\geq 4^{-1 / |F_v| \cdot \sum_{x \in F_v} w_x(v)}$$

This is f(average(w_x))

Extending an Existing Coloring

- **Theorem:** The probability that a node $v \in V_U$ can keep its random color is at least 1/4
- P(v can keep its color X_v)

$$= \sum_{x \in F_v} 1 / |F_v| \cdot P(\, v \text{ can keep color } x \mid X_v = x \,)$$

$$\geq \sum_{x \in F_v} 1 / |F_v| \cdot 4^{-w_x(v)}$$

Let $f(w_x) = 4^{-w_x(v)}$. This is average(f(w_x))

$$\geq 4^{-1 / |F_v| \cdot \sum_{x \in F_v} w_x(v)}$$

This is f(average(w_x))

average(f(x)) ≥ f(average(x)) if f is convex

Extending an Existing Coloring

- **Theorem:** The probability that a node $v \in V_U$ can keep its random color is at least **1/4**
- P(v can keep its color X_v)

$$= \sum_{x \in F_v} 1 / |F_v| \cdot P(v \text{ can keep color } x \mid X_v = x)$$

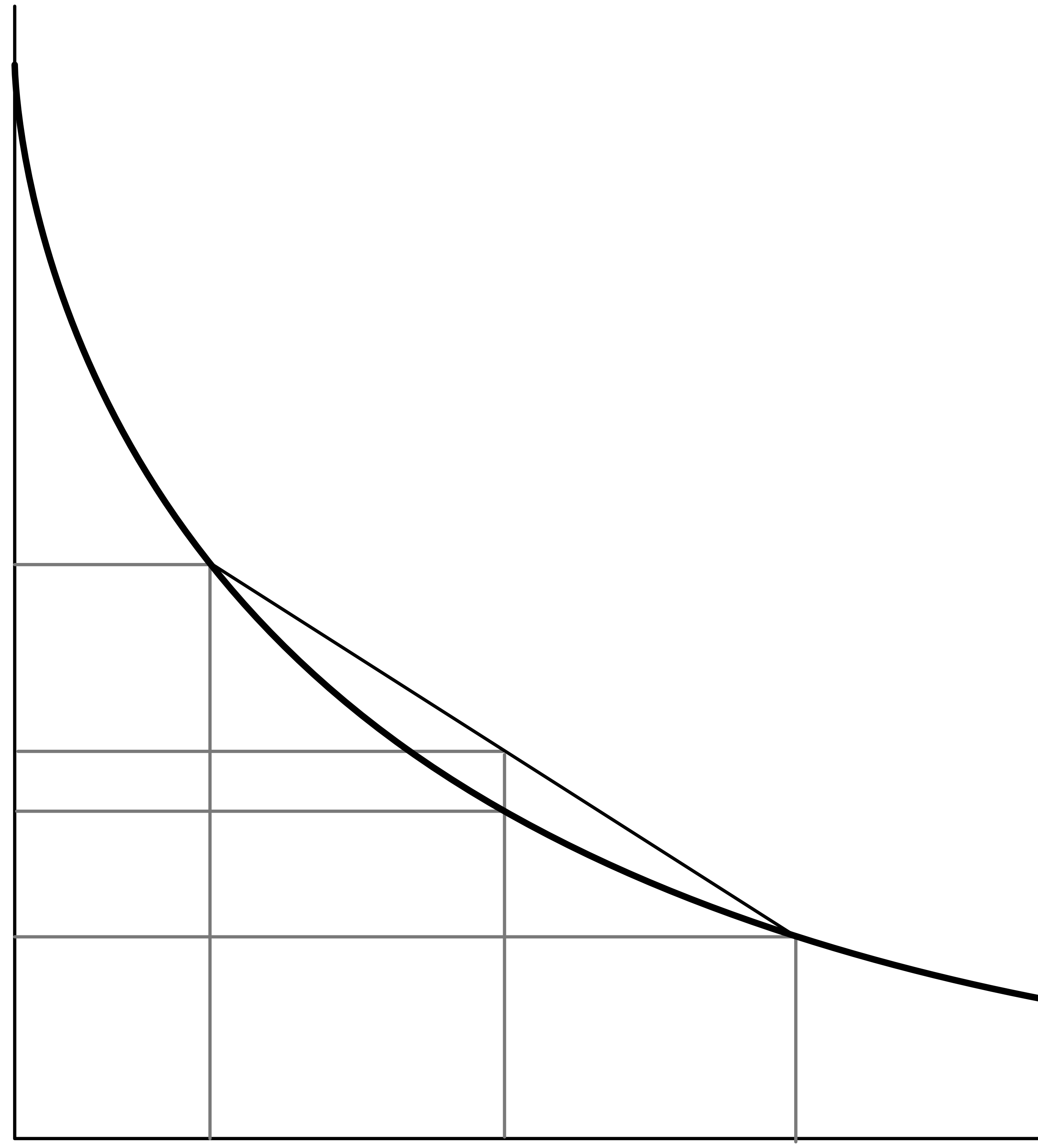
$$\geq \sum_{x \in F_v} 1 / |F_v| \cdot 4^{- w_x(v)}$$

Let $f(w_x) = 4^{-w_x(v)}$. This is average($f(w_x)$)

$$\geq 4^{- 1 / |F_v| \cdot \sum_{x \in F_v} w_x(v)}$$

This is f(average(w_x))

average(f(x)) ≥ f(average(x)) if f is **convex**



Extending an Existing Coloring

- **Theorem:** The probability that a node $v \in V_U$ can keep its random color is at least 1/4
- P(v can keep its color X_v)

$$= \sum_{x \in F_v} 1 / |F_v| \cdot P(\, v \text{ can keep color } x \mid X_v = x \,)$$

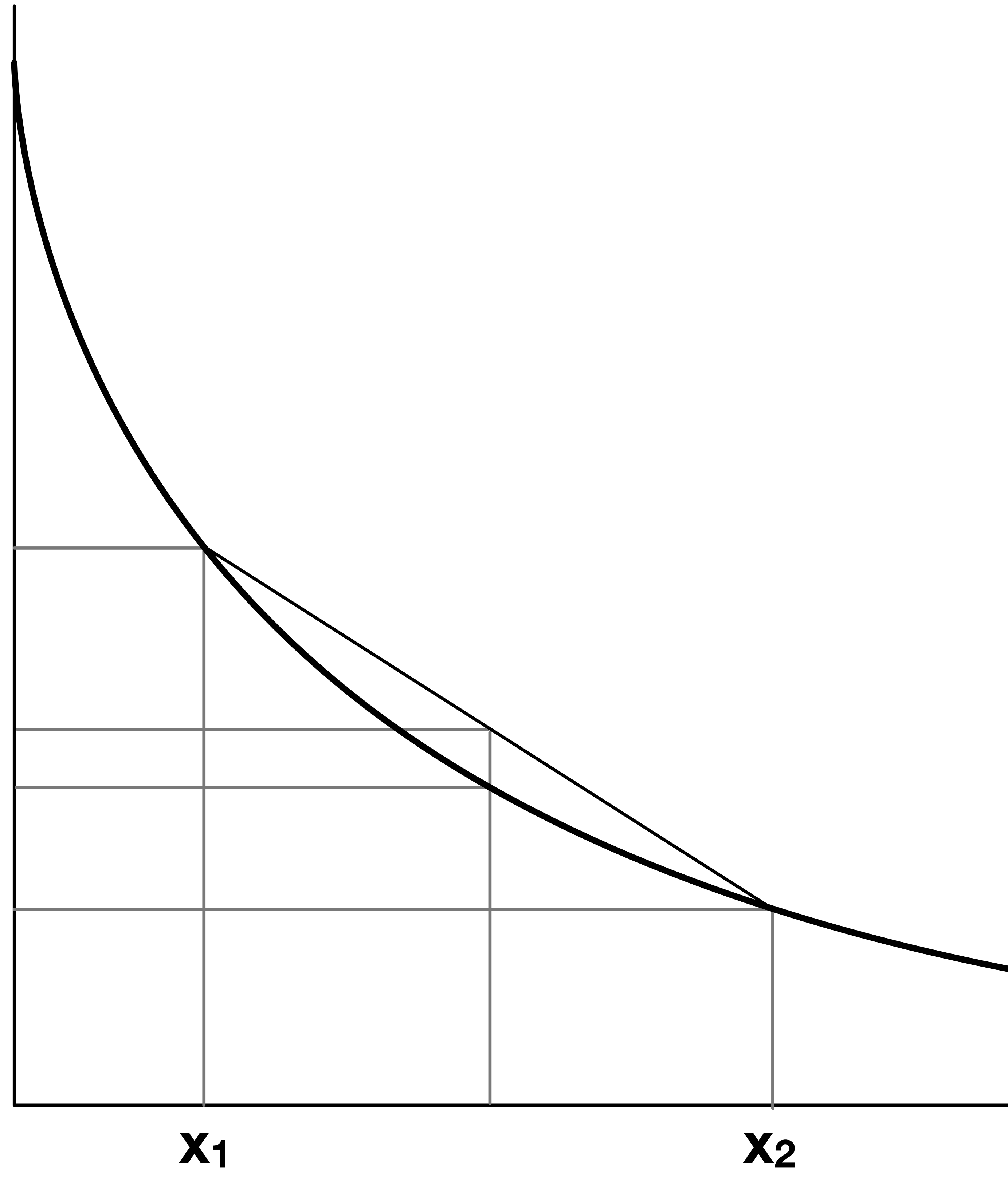
$$\geq \sum_{x \in F_v} 1 / |F_v| \cdot 4^{-w_x(v)}$$

Let $f(w_x) = 4^{-w_x(v)}$. This is average(f(w_x))

$$\geq 4^{-1 / |F_v| \cdot \sum_{x \in F_v} w_x(v)}$$

This is f(average(w_x))

average(f(x)) ≥ f(average(x)) if f is convex



Extending an Existing Coloring

- **Theorem:** The probability that a node $v \in V_U$ can keep its random color is at least 1/4
- P(v can keep its color X_v)

$$= \sum_{x \in F_v} 1 / |F_v| \cdot P(\text{ v can keep color } x \mid X_v = x)$$

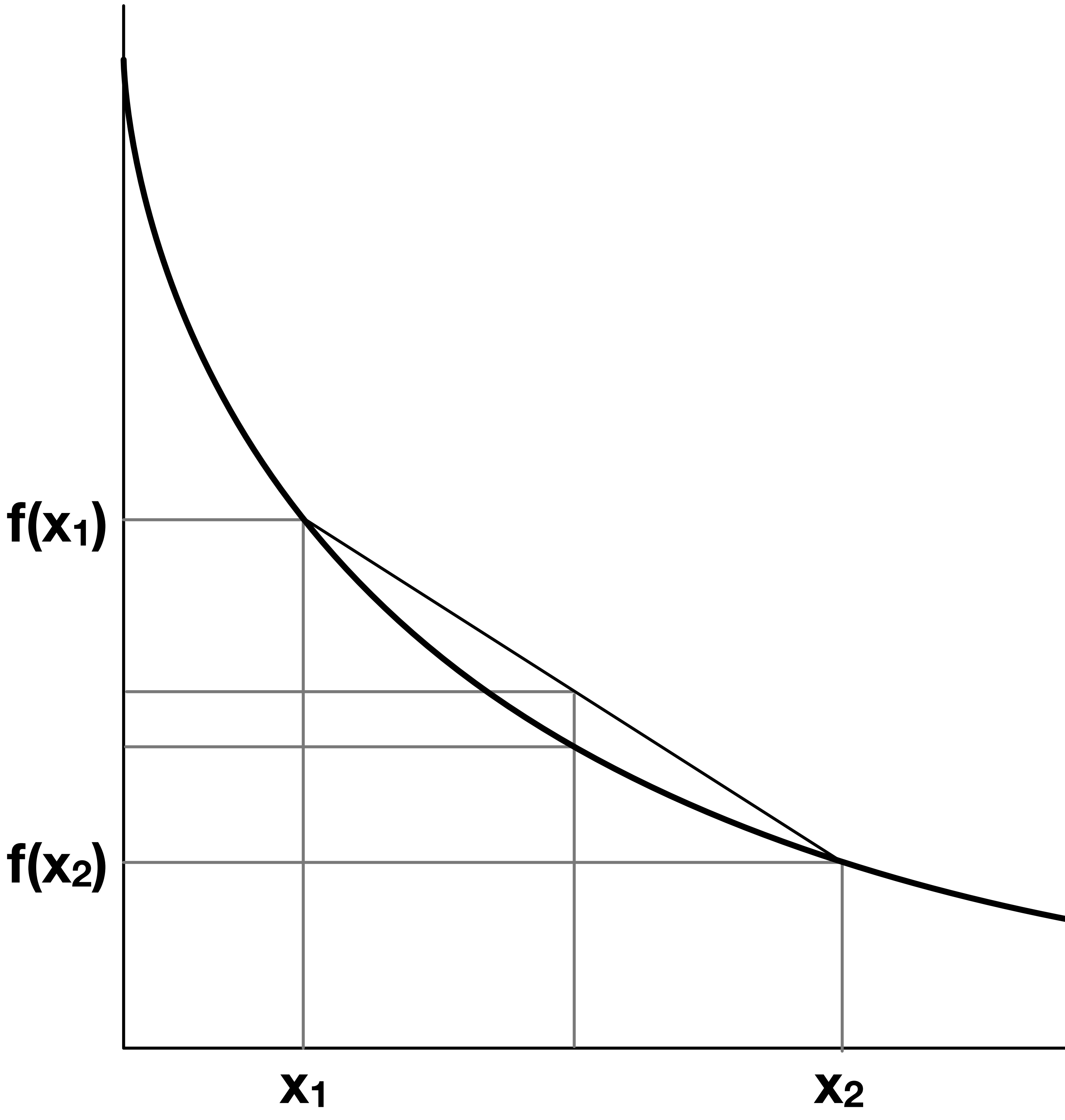
$$\geq \sum_{x \in F_v} 1 / |F_v| \cdot 4^{-w_x(v)}$$

Let $f(w_x) = 4^{-w_x(v)}$. This is average(f(wx))

$$\geq 4^{-1 / |F_v| \cdot \sum_{x \in F_v} w_x(v)}$$

This is f(average(wx))

average(f(x)) ≥ f(average(x)) if f is convex



Extending an Existing Coloring

- **Theorem:** The probability that a node $v \in V_U$ can keep its random color is at least $1/4$
- $P(v \text{ can keep its color } X_v)$

$$= \sum_{x \in F_v} 1 / |F_v| \cdot P(v \text{ can keep color } x \mid X_v = x)$$

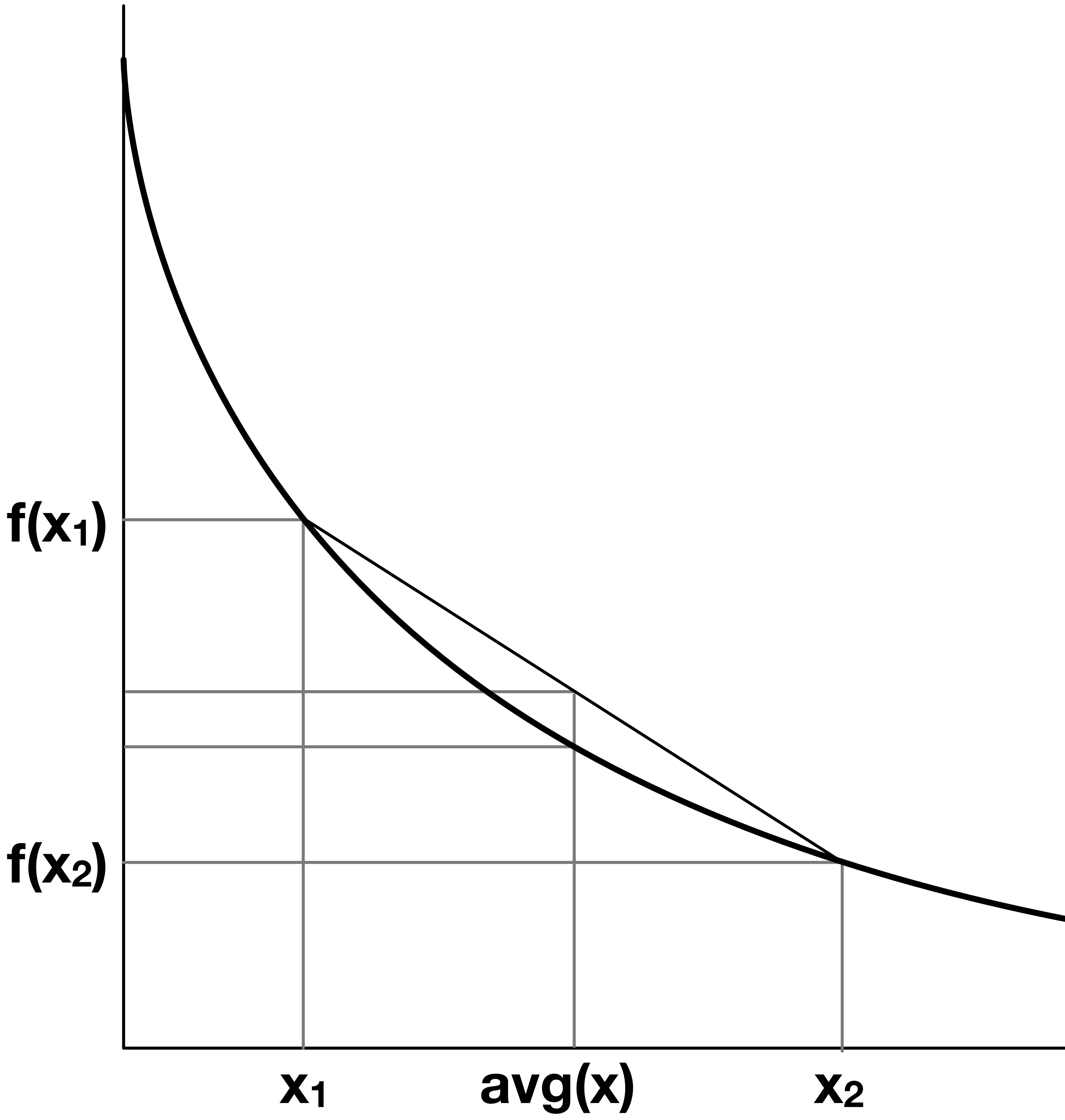
$$\geq \sum_{x \in F_v} 1 / |F_v| \cdot 4^{-w_x(v)}$$

Let $f(w_x) = 4^{-w_x(v)}$. This is average($f(w_x)$)

$$\geq 4^{-1/|F_v| \cdot \sum_{x \in F_v} w_x(v)}$$

This is $f(\text{average}(w_x))$

average(f(x)) ≥ f(average(x)) if f is convex



Extending an Existing Coloring

- **Theorem:** The probability that a node $v \in V_U$ can keep its random color is at least $1/4$
- P(v can keep its color X_v)

$$= \sum_{x \in F_v} 1 / |F_v| \cdot P(\, v \text{ can keep color } x \mid X_v = x \,)$$

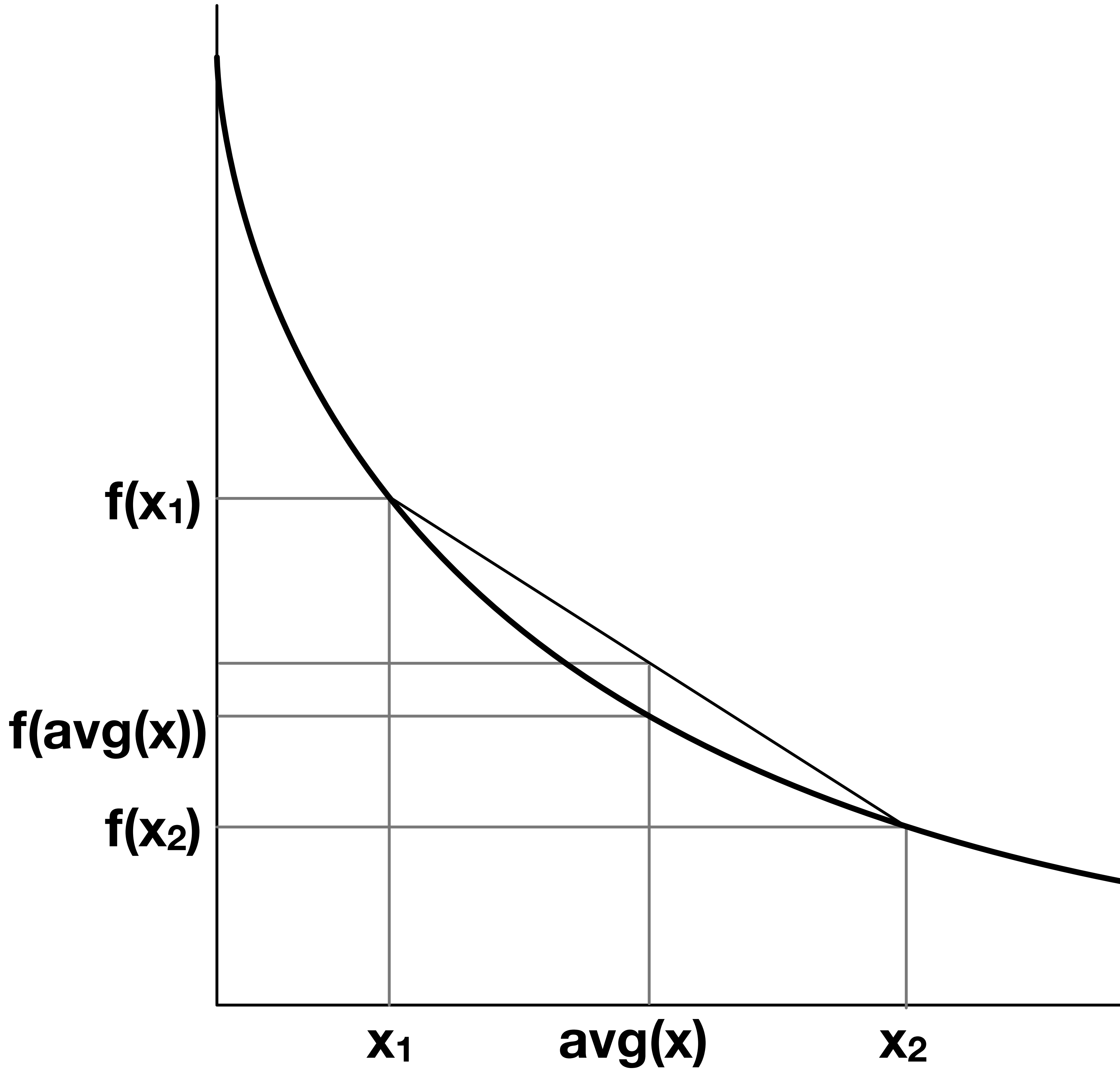
$$\geq \sum_{x \in F_v} 1 / |F_v| \cdot 4^{-w_x(v)}$$

Let $f(w_x) = 4^{-w_x(v)}$. This is average($f(w_x)$)

$$\geq 4^{-1 / |F_v| \cdot \sum_{x \in F_v} w_x(v)}$$

This is f(average(w_x))

average(f(x)) ≥ f(average(x)) if f is convex



Extending an Existing Coloring

- **Theorem:** The probability that a node $v \in V_U$ can keep its random color is at least 1/4
- P(v can keep its color X_v)

$$= \sum_{x \in F_v} 1 / |F_v| \cdot P(\, v \text{ can keep color } x \mid X_v = x \,)$$

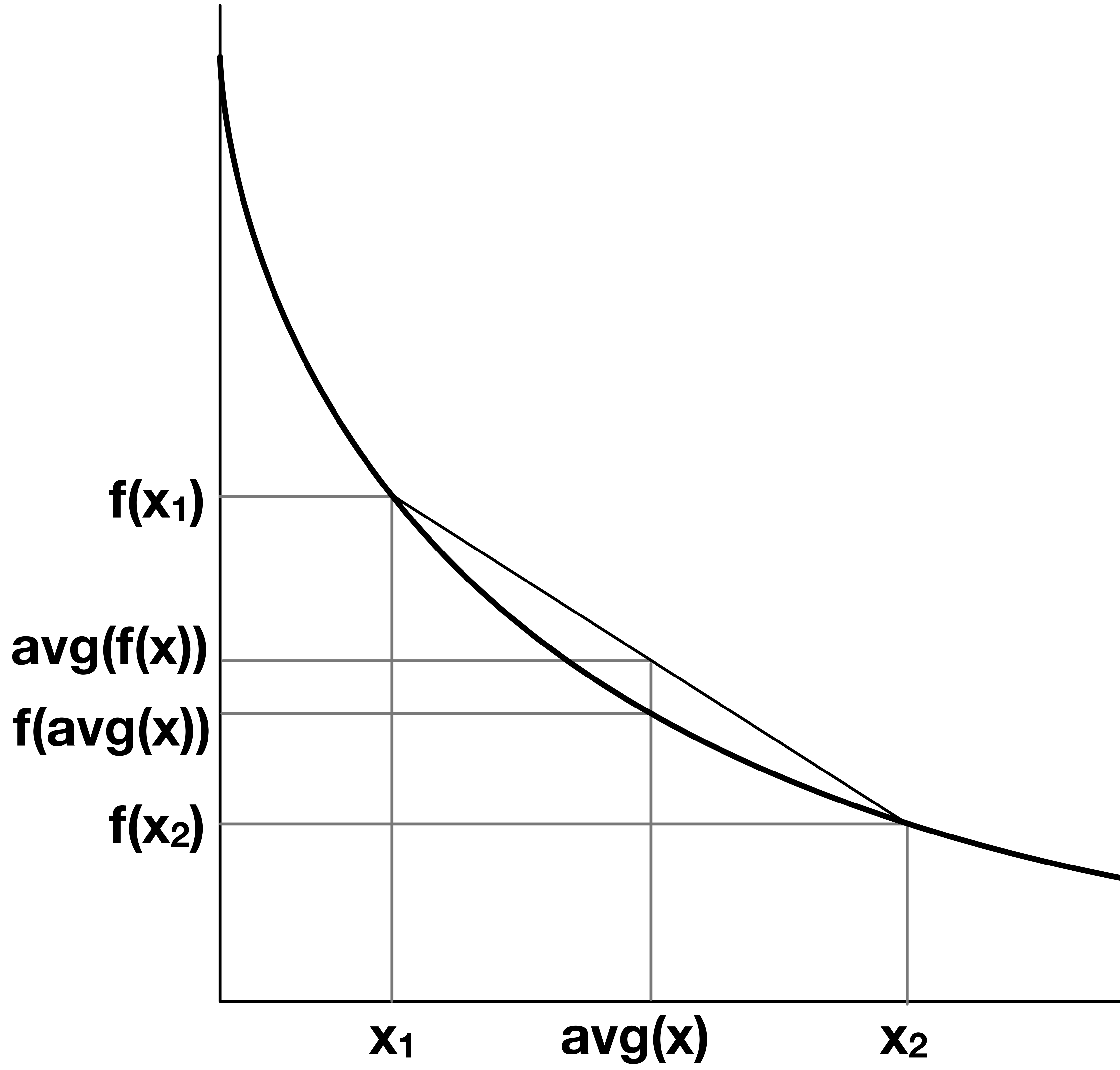
$$\geq \sum_{x \in F_v} 1 / |F_v| \cdot 4^{-w_x(v)}$$

Let $f(w_x) = 4^{-w_x(v)}$. This is average($f(w_x)$)

$$\geq 4^{-1 / |F_v| \cdot \sum_{x \in F_v} w_x(v)}$$

This is f(average(w_x))

average($f(x)$) ≥ f(average(x)) if f is convex



Extending an Existing Coloring

- **Theorem:** The probability that a node $v \in V_U$ can keep its random color is at least $1/4$
- P(v can keep its color X_v)

$$= \sum_{x \in F_v} 1 / |F_v| \cdot P(\, v \text{ can keep color } x \mid X_v = x \,)$$

$$\geq \sum_{x \in F_v} 1 / |F_v| \cdot 4^{-w_x(v)}$$

$$\geq 4^{-1 / |F_v| \cdot \sum_{x \in F_v} w_x(v)}$$

Extending an Existing Coloring

- **Theorem:** The probability that a node $v \in V_U$ can keep its random color is at least $1/4$
- P(v can keep its color X_v)

$$= \sum_{x \in F_v} 1 / |F_v| \cdot P(\, v \text{ can keep color } x \mid X_v = x \,)$$

$$\geq \sum_{x \in F_v} 1 / |F_v| \cdot 4^{-w_x(v)}$$

$$\geq 4^{-1 / |F_v| \cdot \sum_{x \in F_v} w_x(v)}$$

$$\geq 4^{-(1 / |F_v|) \cdot (|F_v| - 1)}$$

Extending an Existing Coloring

- **Theorem:** The probability that a node $v \in V_U$ can keep its random color is at least $1/4$
- P(v can keep its color X_v)

$$= \sum_{x \in F_v} 1 / |F_v| \cdot P(\, v \text{ can keep color } x \mid X_v = x\,)$$

$$\geq \sum_{x \in F_v} 1 / |F_v| \cdot 4^{-w_x(v)}$$

$$\geq 4^{-1 / |F_v| \cdot \sum_{x \in F_v} w_x(v)}$$

$$\geq 4^{-(1 / |F_v|) \cdot (|F_v| - 1)}$$

$$\geq 4^{-1}$$

Randomized Coloring

Randomized Coloring

- **Theorem:** the discussed randomized coloring algorithm computes a valid coloring of **$G=(V, E)$** in **$O(\log n)$** rounds in expectation and **with high probability**. Every node **$v \in V$** gets a color in **$\{1, \ldots, \deg(v) + 1\}$**

Randomized Coloring

- **Theorem:** the discussed randomized coloring algorithm computes a valid coloring of **$G=(V, E)$** in **$O(\log n)$** rounds in expectation and **with high probability**. Every node **$v \in V$** gets a color in **$\{1, \ldots, \deg(v) + 1\}$**

- **P(** v uncolored after **T** phases **)** $\leq$ **$(3/4)^T$**

Randomized Coloring

- **Theorem:** the discussed randomized coloring algorithm computes a valid coloring of $G=(V, E)$ in $O(\log n)$ rounds in expectation and **with high probability**. Every node $v \in V$ gets a color in $\{1, \ldots, deg(v) + 1\}$

- **P(** v uncolored after **T** phases **)** $\leq (3/4)^T$

- Choose $T = (c+1) \log_{4/3} n = O(\log n)$

Randomized Coloring

- **Theorem:** the discussed randomized coloring algorithm computes a valid coloring of $G=(V, E)$ in $O(\log n)$ rounds in expectation and **with high probability**. Every node $v \in V$ gets a color in $\{1, \ldots, \deg(v) + 1\}$
- **P(** v uncolored after **T** phases **) ≤** $(3/4)^T$
- Choose $T = (c+1) \log_{4/3} n = O(\log n)$
- **P(** v uncolored after **T** phases **) ≤** $1 / n^{c+1}$

Randomized Coloring

- **Theorem:** the discussed randomized coloring algorithm computes a valid coloring of $G=(V, E)$ in $O(\log n)$ rounds in expectation and **with high probability**. Every node $v \in V$ gets a color in $\{1, \ldots, \deg(v) + 1\}$

- **P(** v uncolored after **T** phases **)** $\leq (3/4)^T$

- Choose $T = (c+1) \log_{4/3} n = O(\log n)$

- **P(** v uncolored after **T** phases **)** $\leq 1 / n^{c+1}$

- **P(** **some node** uncolored after **T** phases **)** $\leq n \cdot 1 / n^{c+1} = 1 / n^c$

Randomized Coloring

- **Theorem:** the discussed randomized coloring algorithm computes a valid coloring of $G=(V, E)$ in $O(\log n)$ rounds in expectation and **with high probability**. Every node $v \in V$ gets a color in $\{1, \ldots, \deg(v) + 1\}$
- $P($ v uncolored after **T** phases $) \leq (3/4)^T$
- Choose $T = (c+1) \log_{4/3} n = O(\log n)$
- $P($ v uncolored after **T** phases $) \leq 1 / n^{c+1}$
- $P($ **some node** uncolored after **T** phases $) \leq n \cdot 1 / n^{c+1} = 1 / n^c$

Union Bound: $P(A \cup B) = P(A) + P(B) - P(A \cap B) \leq P(A) + P(B)$

Randomized MIS: Ideas

Randomized MIS: Ideas

- Compute a **(Δ+1)**-coloring, then **convert** it into an **MIS** as seen in the last lecture:

Randomized MIS: Ideas

- Compute a **(Δ+1)**-coloring, then **convert** it into an **MIS** as seen in the last lecture:
  - requires **O(Δ + log n)**

Randomized MIS: Ideas

- Compute a **(Δ+1)**-coloring, then **convert** it into an **MIS** as seen in the last lecture:
    - requires **O(Δ + log n)**
- Compute a **random ID assignment**, then run the **greedy** algorithm seen in the last lecture:

Randomized MIS: Ideas

- Compute a **(Δ+1)**-coloring, then **convert** it into an **MIS** as seen in the last lecture:
  - requires **O(Δ + log n)**
- Compute a **random ID assignment**, then run the **greedy** algorithm seen in the last lecture:
  - requires **O(log n)**, but its analysis is highly non-trivial

Randomized MIS: Ideas

- Compute a **(Δ+1)**-coloring, then **convert** it into an **MIS** as seen in the last lecture:
  - requires **O(Δ + log n)**
- Compute a **random ID assignment**, then run the **greedy** algorithm seen in the last lecture:
  - requires **O(log n)**, but its analysis is highly non-trivial
- We are going to follow a similar approach. At each round:

Randomized MIS: Ideas

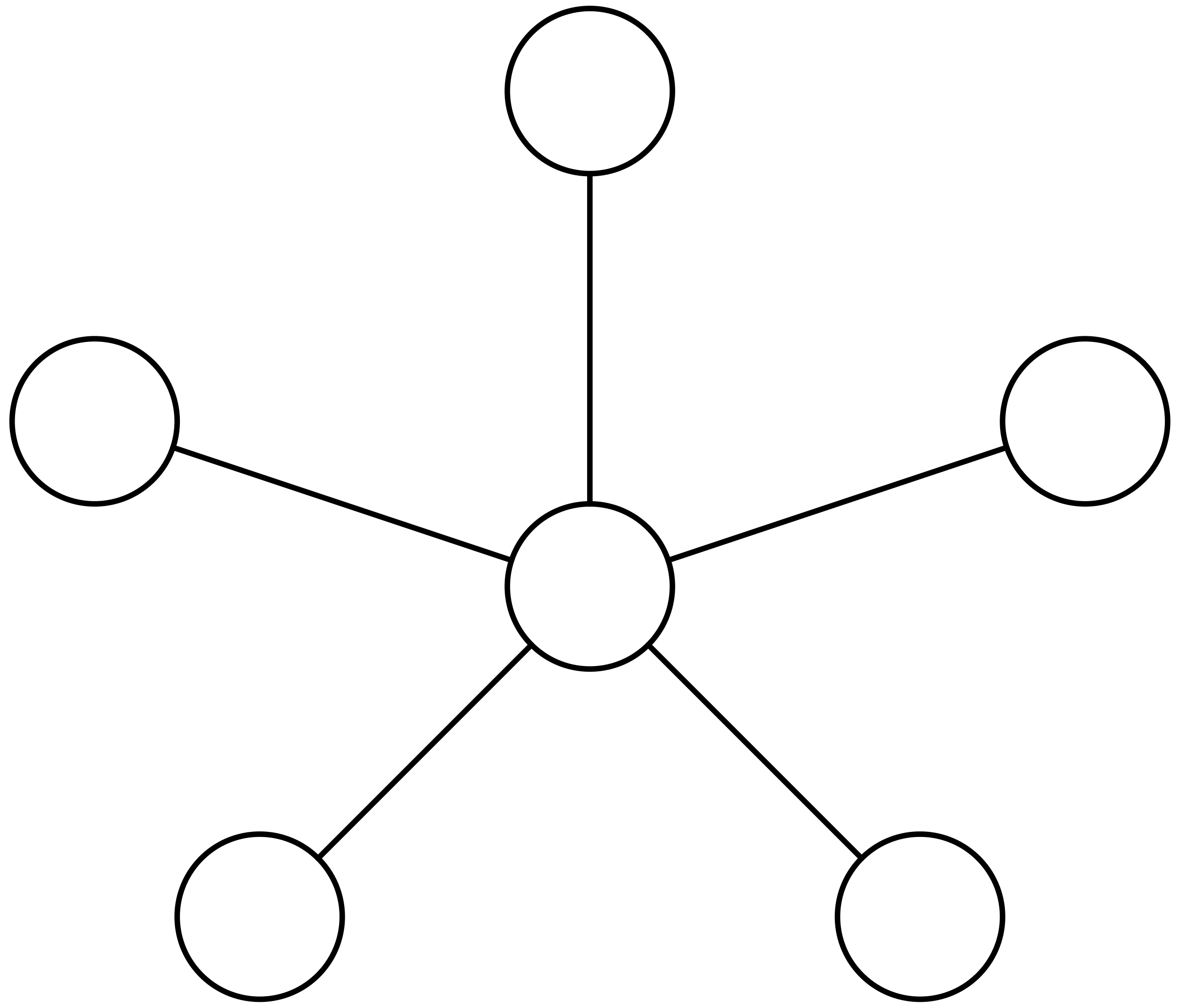
- Compute a **(Δ+1)**-coloring, then **convert** it into an **MIS** as seen in the last lecture:
    - requires **O(Δ + log n)**
- Compute a **random ID assignment**, then run the **greedy** algorithm seen in the last lecture:
    - requires **O(log n)**, but its analysis is highly non-trivial
- We are going to follow a similar approach. At each round:
    - Each node v picks a random number $R_v \in [0,1]$

Randomized MIS: Ideas

- Compute a **$(\Delta+1)$**-coloring, then **convert** it into an **MIS** as seen in the last lecture:
  - requires **$O(\Delta + \log n)$**
- Compute a **random ID assignment**, then run the **greedy** algorithm seen in the last lecture:
  - requires **$O(\log n)$**, but its analysis is highly non-trivial
- We are going to follow a similar approach. At each round:
  - Each node v picks a random number **$R_v \in [0,1]$**
  - Node **v** joins the MIS if **$R_v < R_u$** for all **$u \in N(v)$**

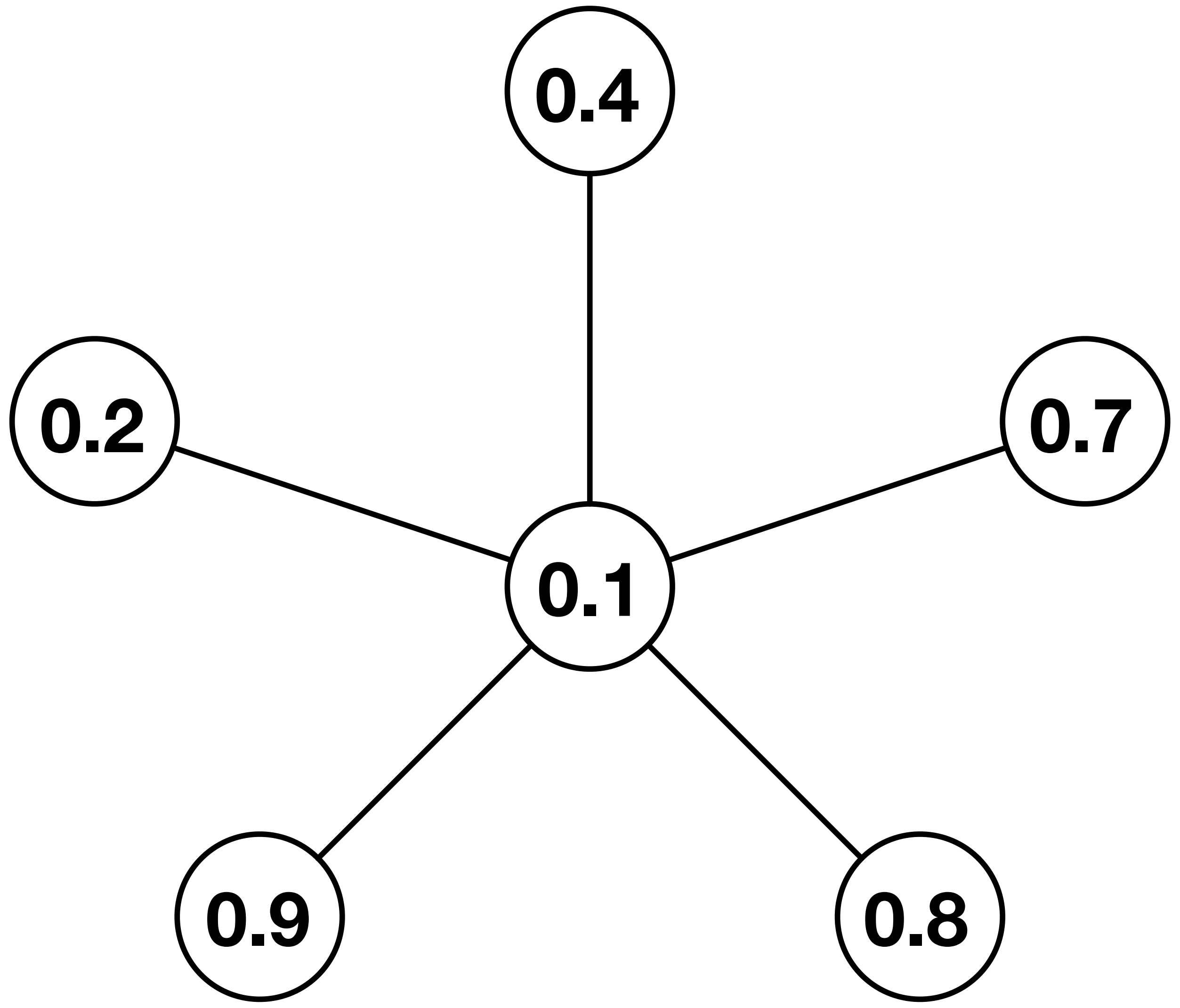
Randomized MIS: Ideas

- Compute a **$(\Delta+1)$**-coloring, then **convert** it into an **MIS** as seen in the last lecture:
  - requires **$O(\Delta + \log n)$**
- Compute a **random ID assignment**, then run the **greedy** algorithm seen in the last lecture:
  - requires **$O(\log n)$**, but its analysis is highly non-trivial
- We are going to follow a similar approach. At each round:
  - Each node v picks a random number **$R_v \in [0,1]$**
  - Node **v** joins the MIS if **$R_v < R_u$** for all **$u \in N(v)$**



Randomized MIS: Ideas

- Compute a $(\Delta+1)$-coloring, then **convert** it into an **MIS** as seen in the last lecture:
  - requires $O(\Delta + \log n)$
- Compute a **random ID assignment**, then run the **greedy** algorithm seen in the last lecture:
  - requires $O(\log n)$, but its analysis is highly non-trivial
- We are going to follow a similar approach. At each round:
  - Each node v picks a random number $R_v \in [0,1]$
  - Node **v** joins the MIS if $R_v < R_u$ for all $u \in N(v)$



Randomized MIS: Luby's Algorithm

Randomized MIS: Luby's Algorithm

- At each round:

Randomized MIS: Luby's Algorithm

- At each round:
  - Each node v picks a random number **$R_v \in [0,1]$**

Randomized MIS: Luby's Algorithm

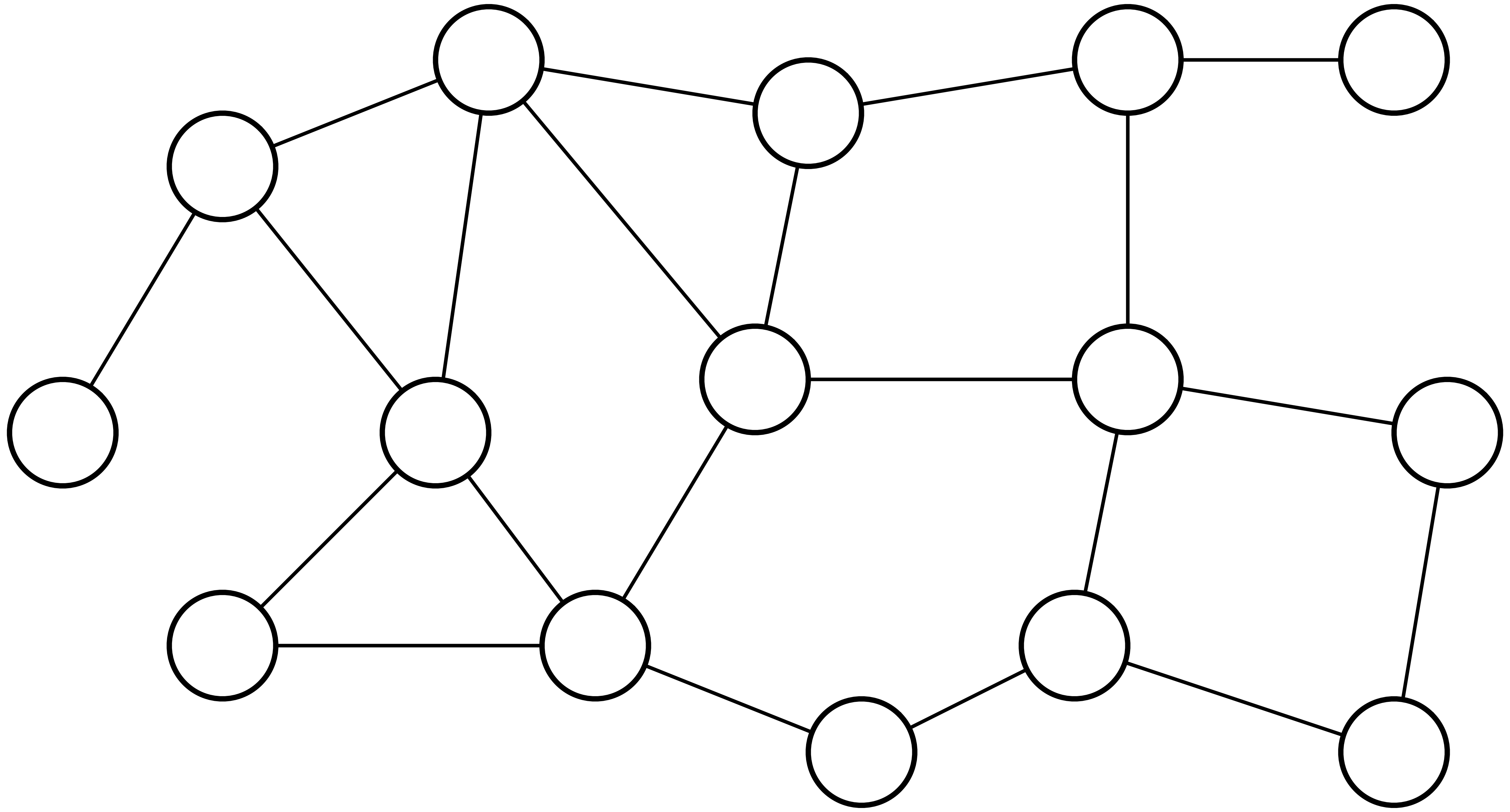
- At each round:
  - Each node v picks a random number **$R_v \in [0,1]$**
  - Node **v** joins the MIS if **$R_v < R_u$** for all **$u \in N(v)$**

Randomized MIS: Luby's Algorithm

- At each round:
  - Each node v picks a random number **$R_v \in [0,1]$**
  - Node **v** joins the MIS if **$R_v < R_u$** for all **$u \in N(v)$**
  - Remove MIS nodes and its neighbors

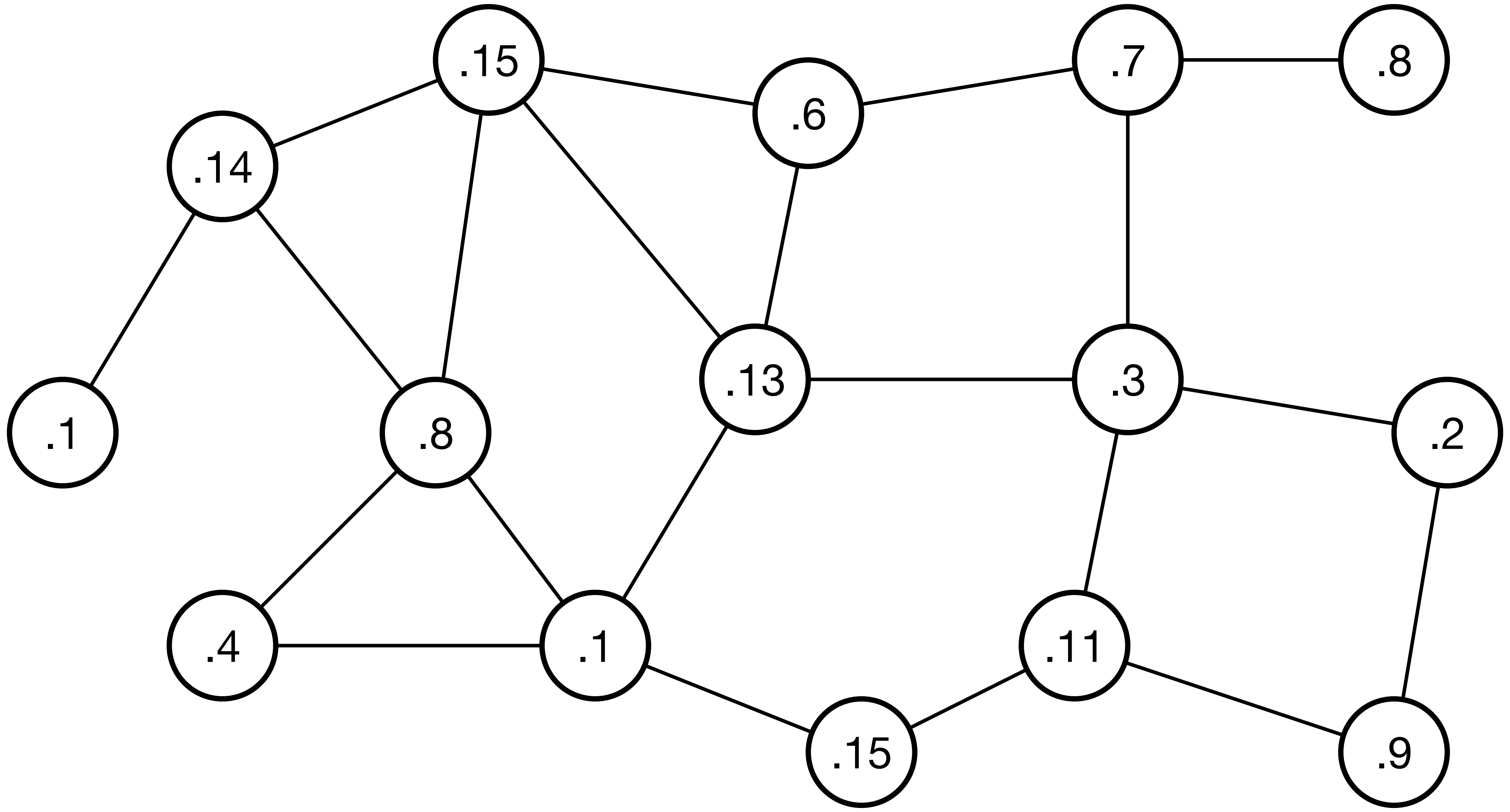
Randomized MIS: Luby's Algorithm

- At each round:
  - Each node v picks a random number **$R_v \in [0,1]$**
  - Node **v** joins the MIS if **$R_v < R_u$** for all **$u \in N(v)$**
  - Remove MIS nodes and its neighbors



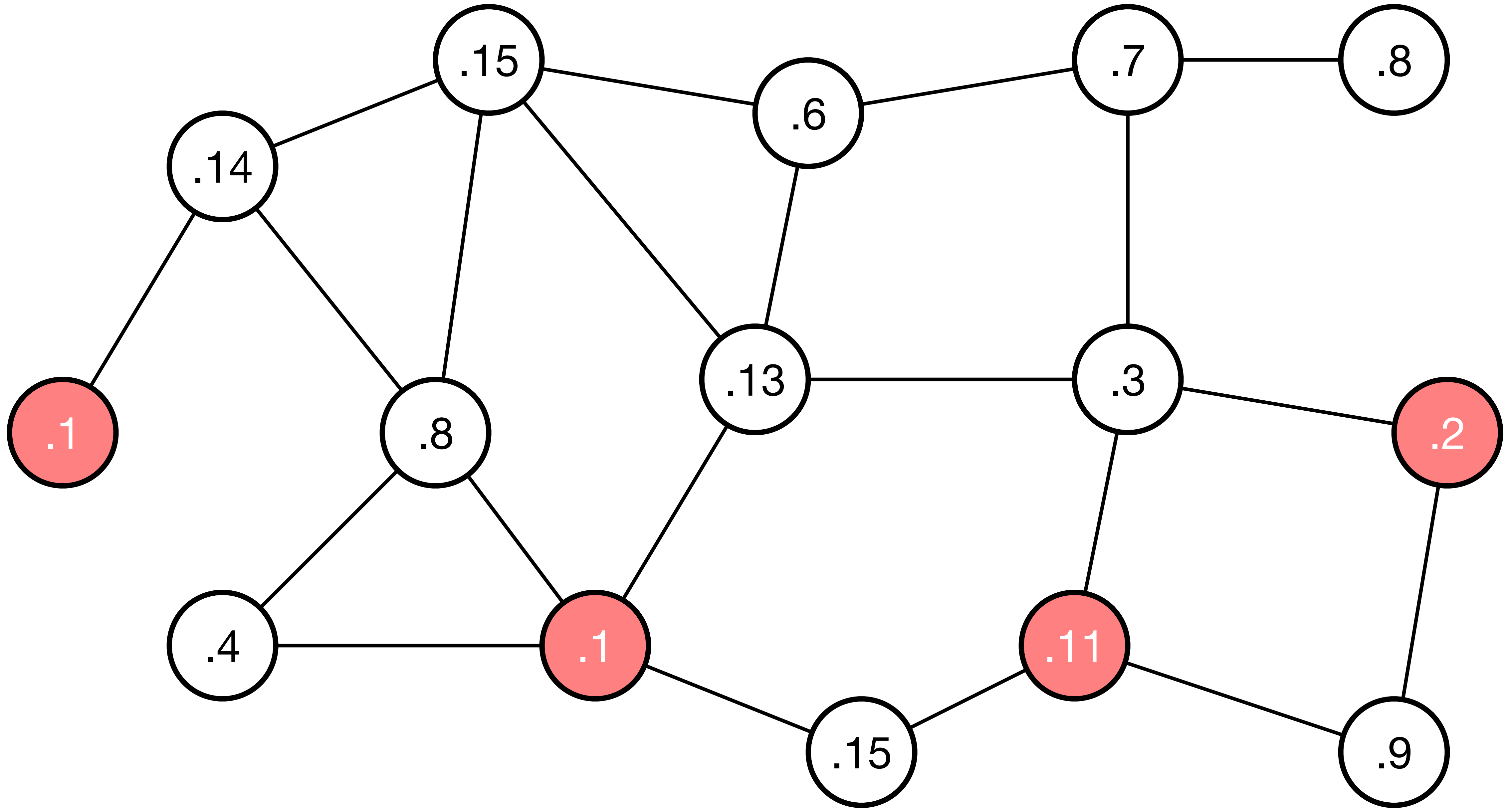
Randomized MIS: Luby's Algorithm

- At each round:
  - Each node v picks a random number $R_v \in [0,1]$
  - Node **v** joins the MIS if $R_v < R_u$ for all $u \in N(v)$
  - Remove MIS nodes and its neighbors



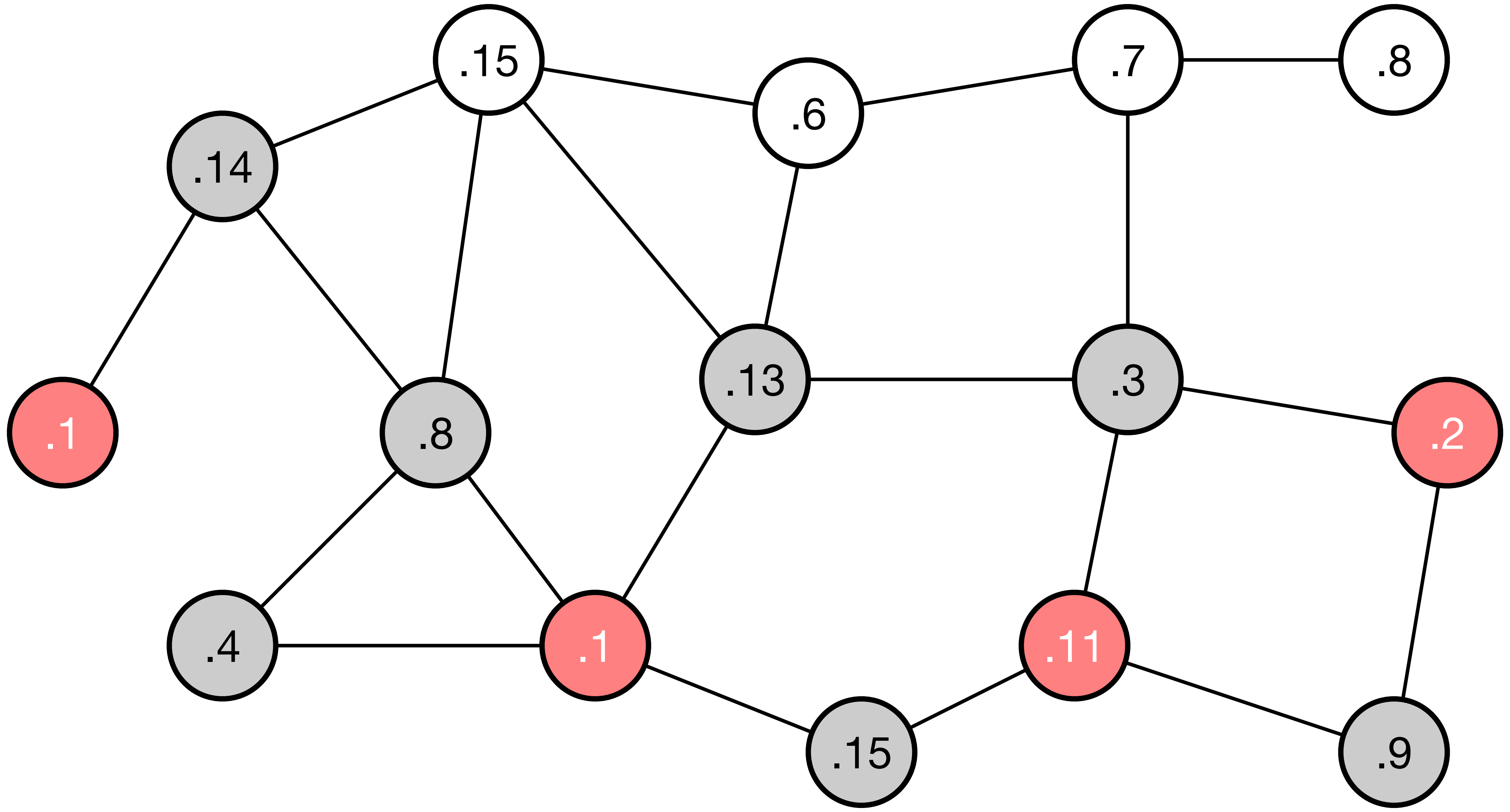
Randomized MIS: Luby's Algorithm

- At each round:
  - Each node v picks a random number **$R_v \in [0,1]$**
  - Node **v** joins the MIS if **$R_v < R_u$** for all **$u \in N(v)$**
  - Remove MIS nodes and its neighbors



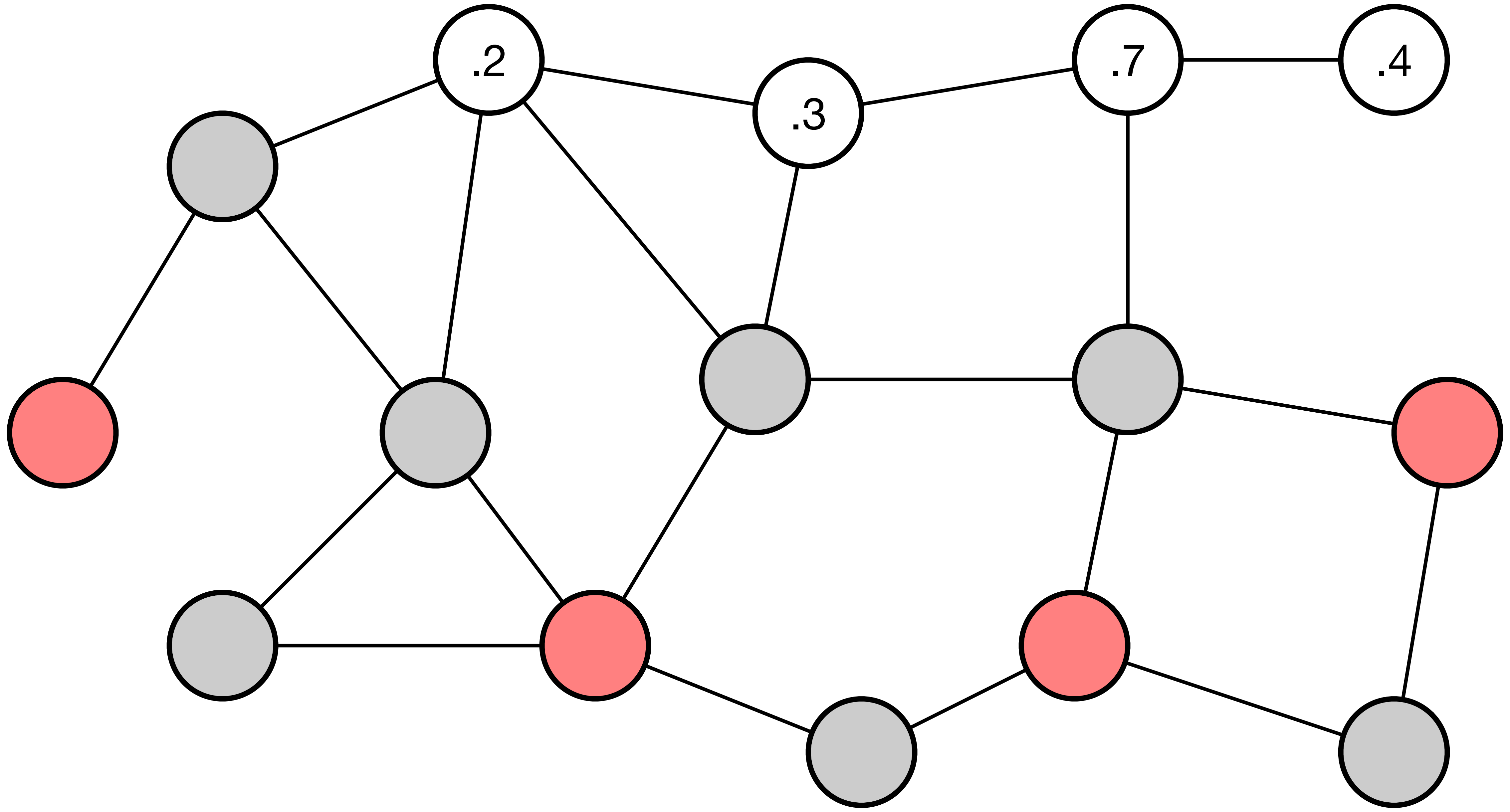
Randomized MIS: Luby's Algorithm

- At each round:
  - Each node v picks a random number **$R_v \in [0,1]$**
  - Node **v** joins the MIS if **$R_v < R_u$** for all **$u \in N(v)$**
  - Remove MIS nodes and its neighbors



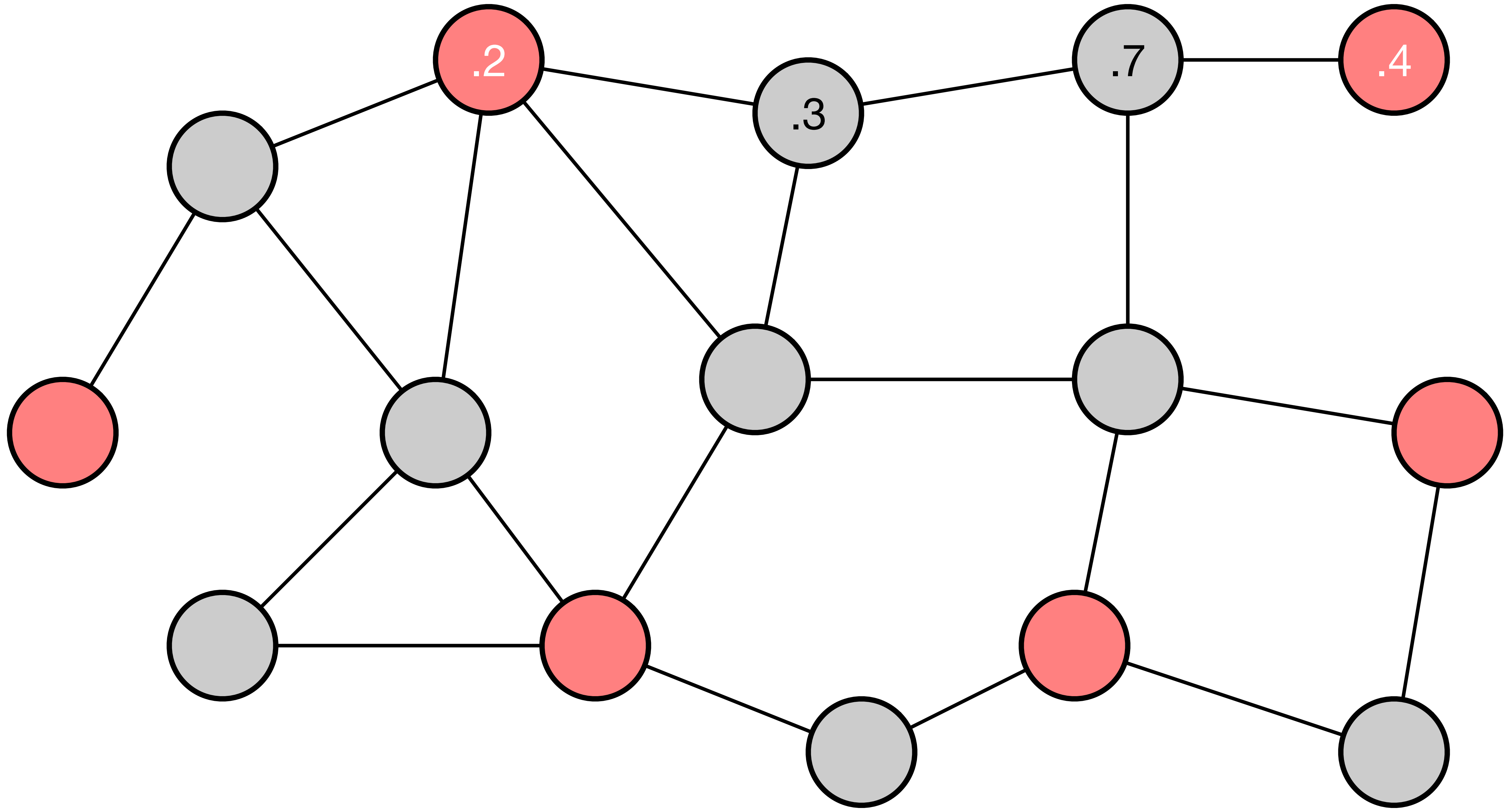
Randomized MIS: Luby's Algorithm

- At each round:
  - Each node v picks a random number $R_v \in [0,1]$
  - Node **v** joins the MIS if $R_v < R_u$ for all $u \in N(v)$
  - Remove MIS nodes and its neighbors



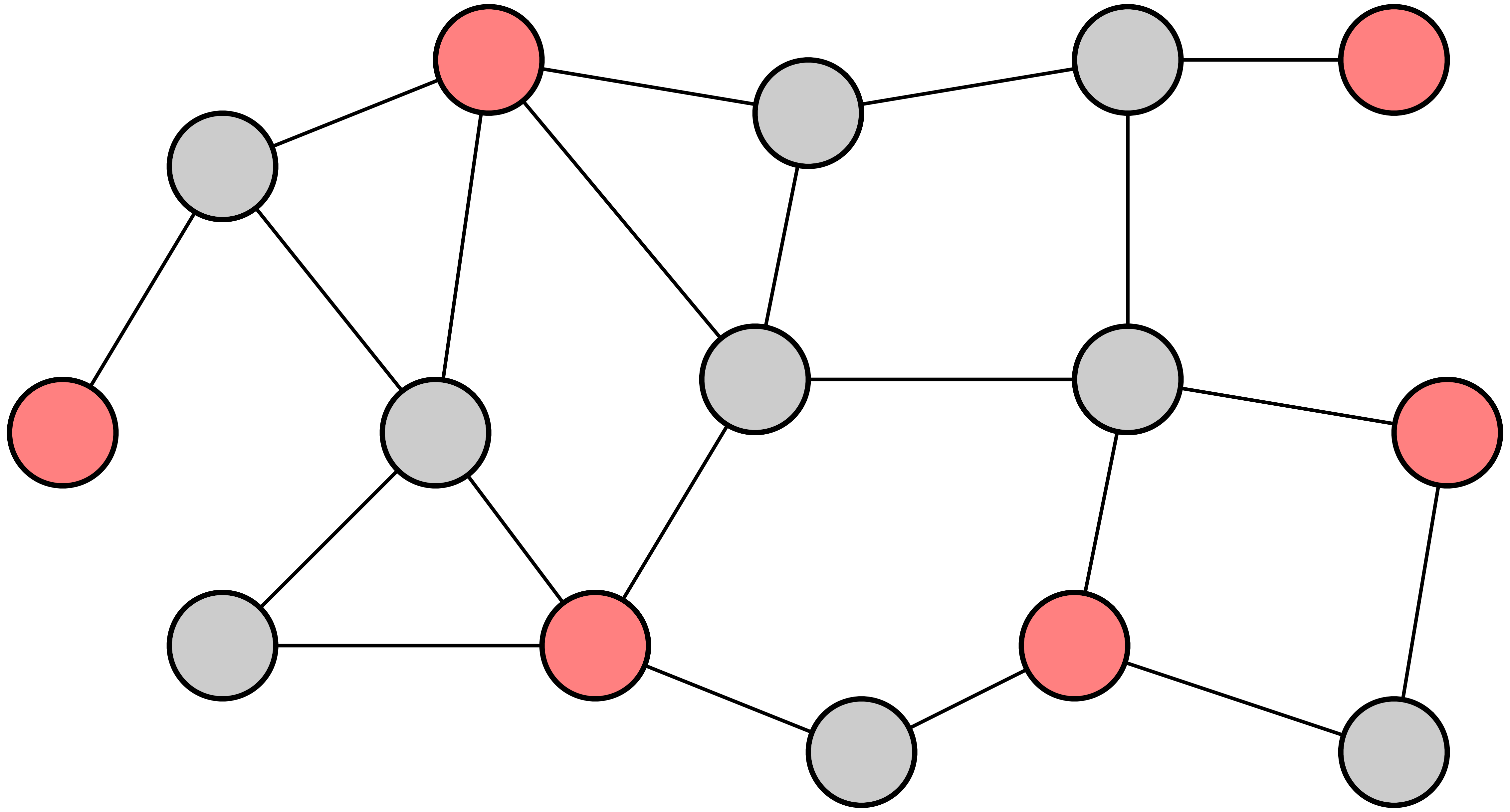
Randomized MIS: Luby's Algorithm

- At each round:
  - Each node v picks a random number $R_v \in [0,1]$
  - Node **v** joins the MIS if $R_v < R_u$ for all $u \in N(v)$
  - Remove MIS nodes and its neighbors



Randomized MIS: Luby's Algorithm

- At each round:
  - Each node v picks a random number $R_v \in [0,1]$
  - Node **v** joins the MIS if $R_v < R_u$ for all $u \in N(v)$
  - Remove MIS nodes and its neighbors



Randomized MIS: Luby's Algorithm

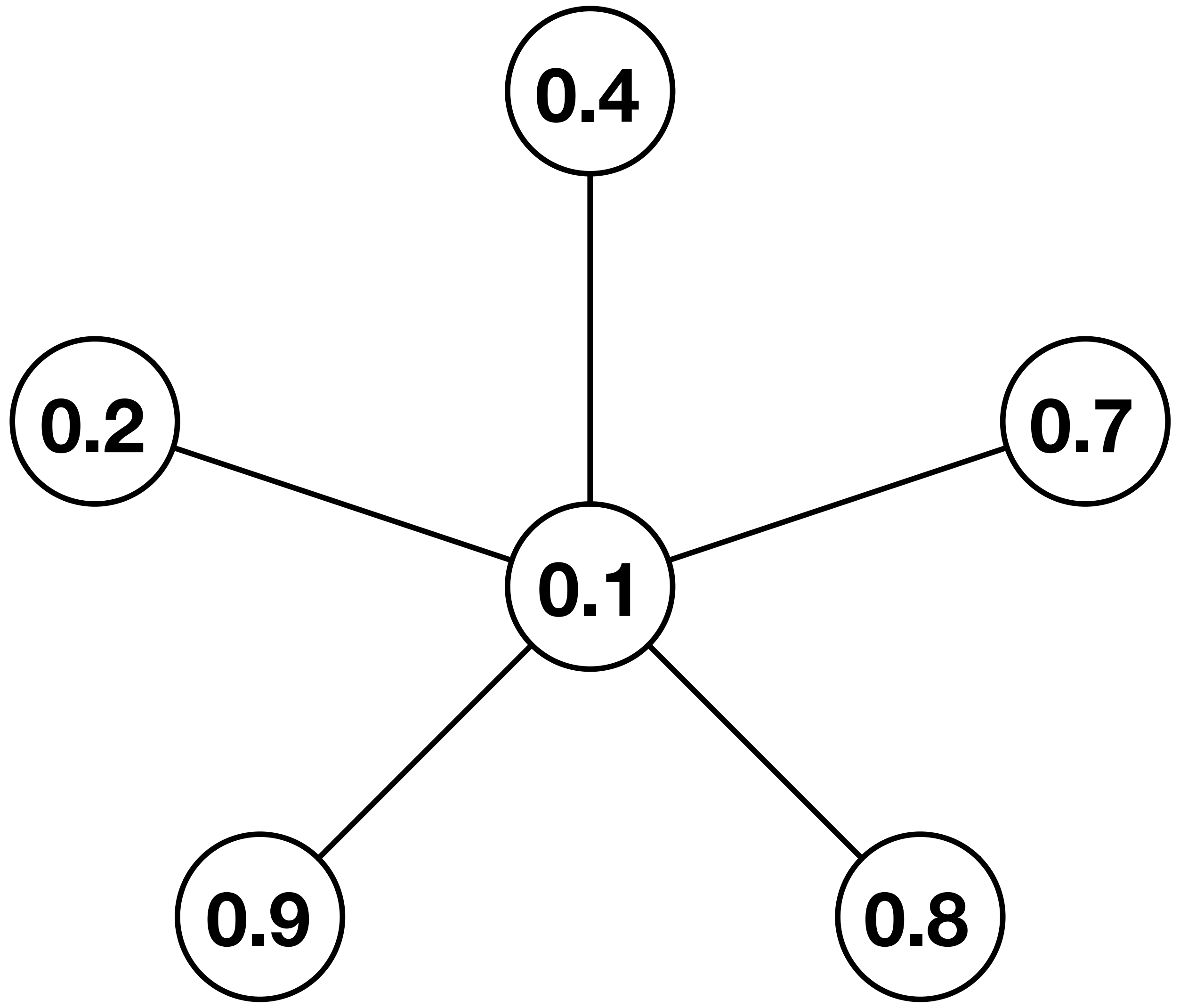
- At each round:
  - Each node v picks a random number $R_v \in [0,1]$
  - Node **v** joins the MIS if $R_v < R_u$ for all $u \in N(v)$
  - Remove MIS nodes and its neighbors

- P(**v** joins **MIS**) = $1 / (deg(v) + 1)$

Randomized MIS: Luby's Algorithm

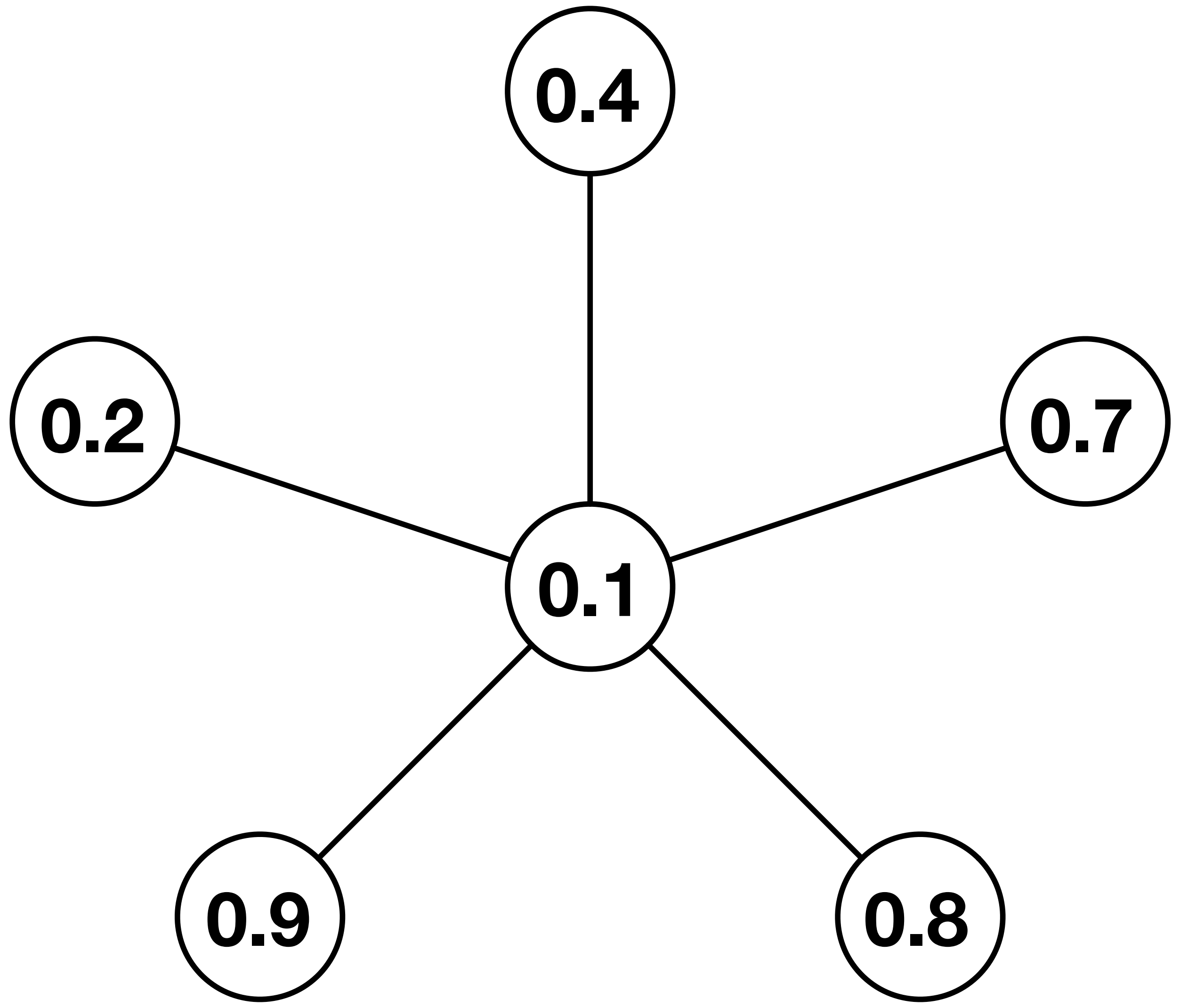
- At each round:
  - Each node v picks a random number **$R_v \in [0,1]$**
  - Node **v** joins the MIS if **$R_v < R_u$** for all **$u \in N(v)$**
  - Remove MIS nodes and its neighbors

- P(**v** joins **MIS**) = **1 / (deg(v) +1)**



Randomized MIS: Luby's Algorithm

- At each round:
  - Each node v picks a random number **$R_v \in [0,1]$**
  - Node **v** joins the MIS if **$R_v < R_u$** for all **$u \in N(v)$**
  - Remove MIS nodes and its neighbors
- P(**v** joins **MIS**) = **1 / (deg(v) +1)**
- Can we prove that a **constant fraction** of nodes gets **removed**?

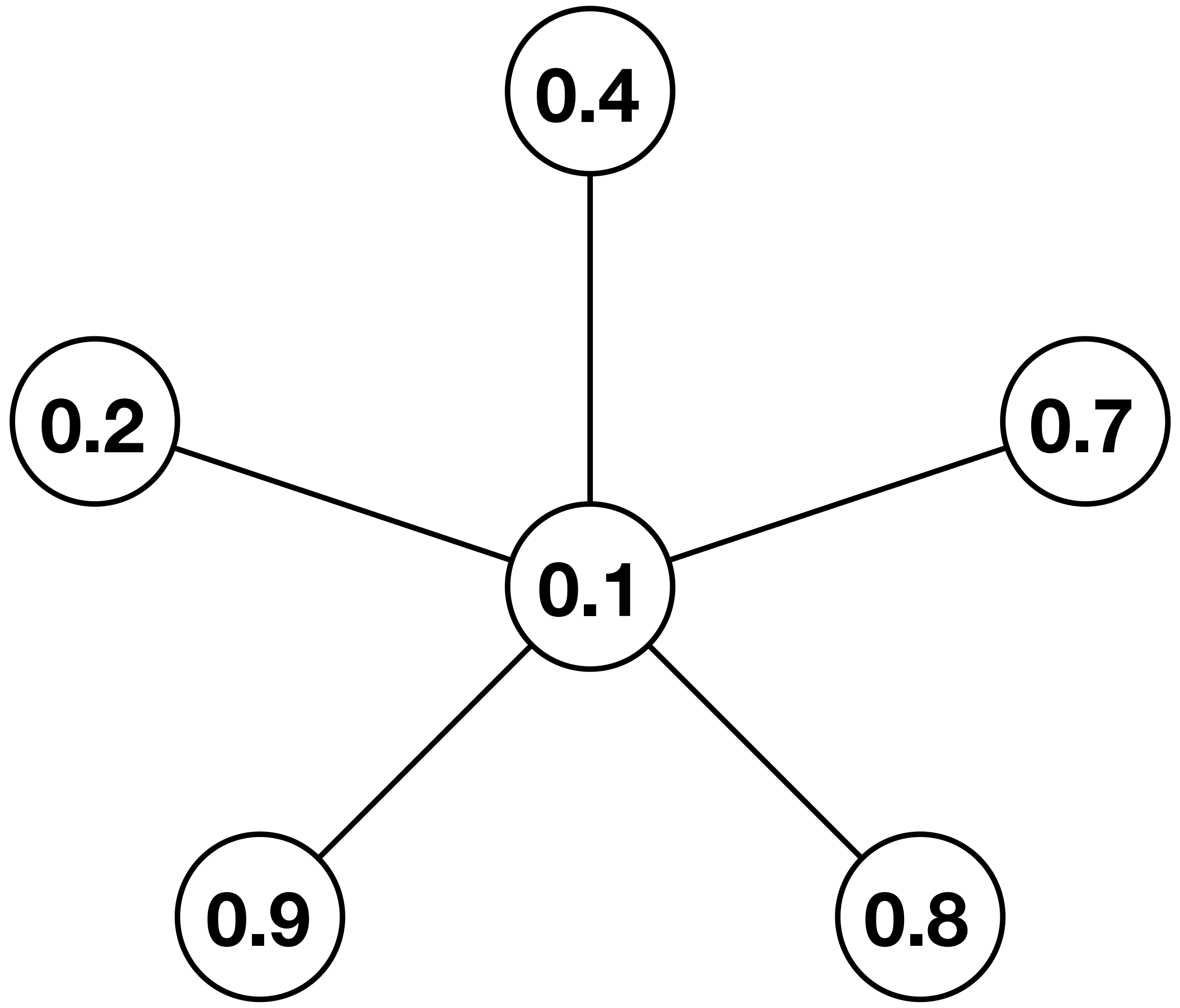


Randomized MIS: Luby's Algorithm

- At each round:
  - Each node v picks a random number **$R_v \in [0,1]$**
  - Node **v** joins the MIS if **$R_v < R_u$** for all **$u \in N(v)$**
  - Remove MIS nodes and its neighbors

- P(**v** joins **MIS**) = **1 / (deg(v) +1)**

- Can we prove that a **constant fraction** of nodes gets **removed**?
  - **No**!



Randomized MIS: Luby's Algorithm

Randomized MIS: Luby's Algorithm

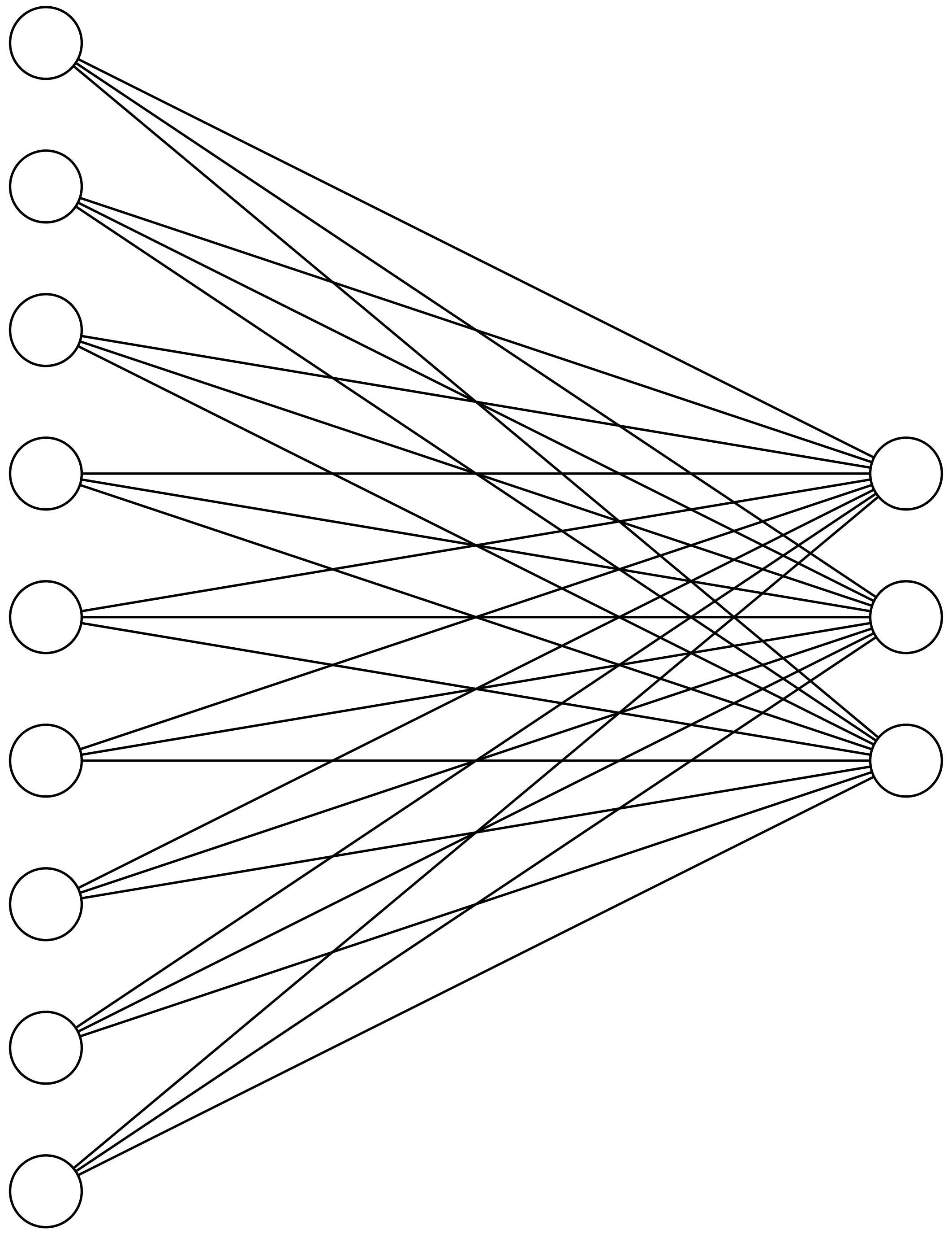
- Can we prove that a **constant fraction** of nodes gets **removed**?

Randomized MIS: Luby's Algorithm

- Can we prove that a **constant fraction** of nodes gets **removed**?
  - **No**!

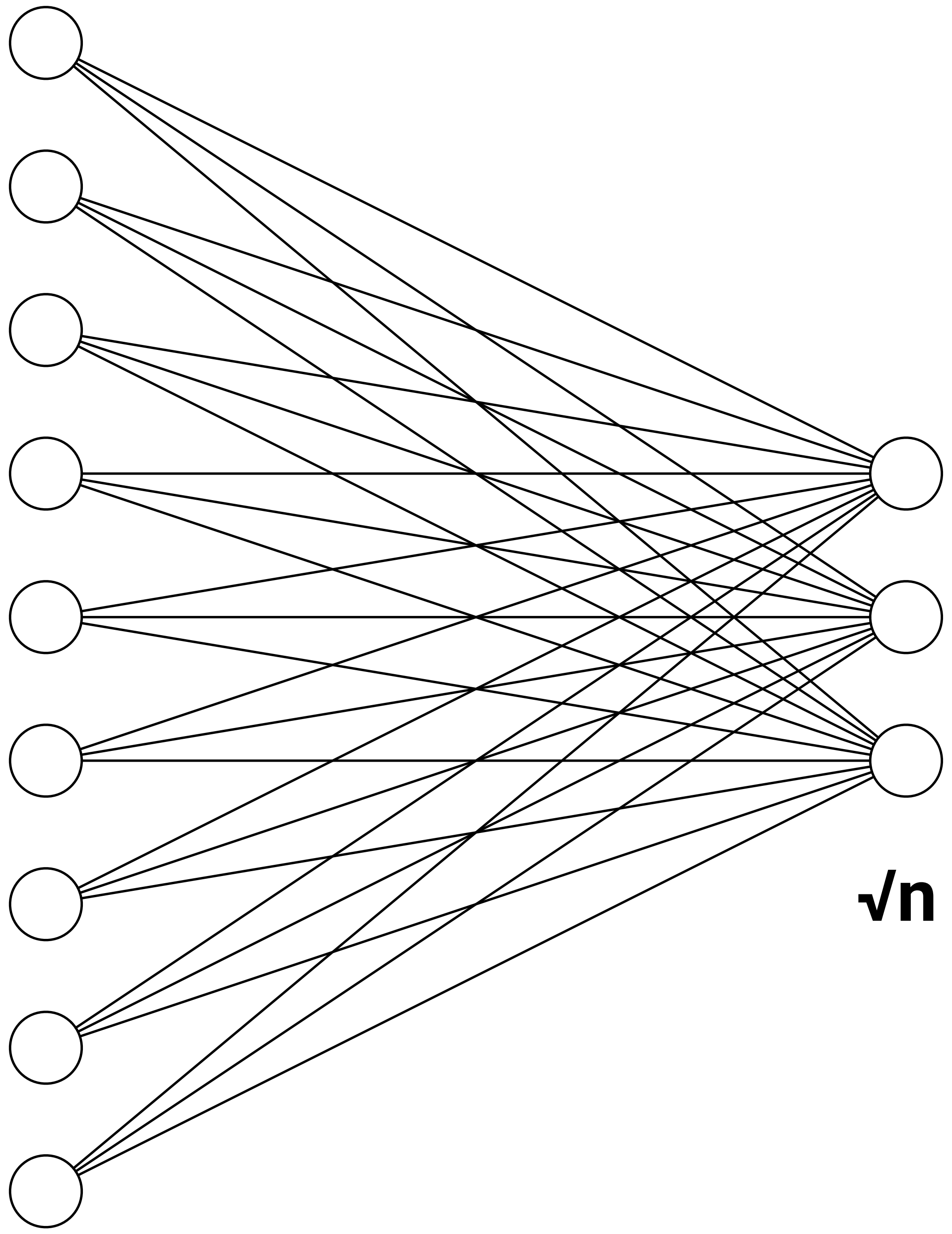
Randomized MIS: Luby's Algorithm

- Can we prove that a **constant fraction** of nodes gets **removed**?
  - **No**!



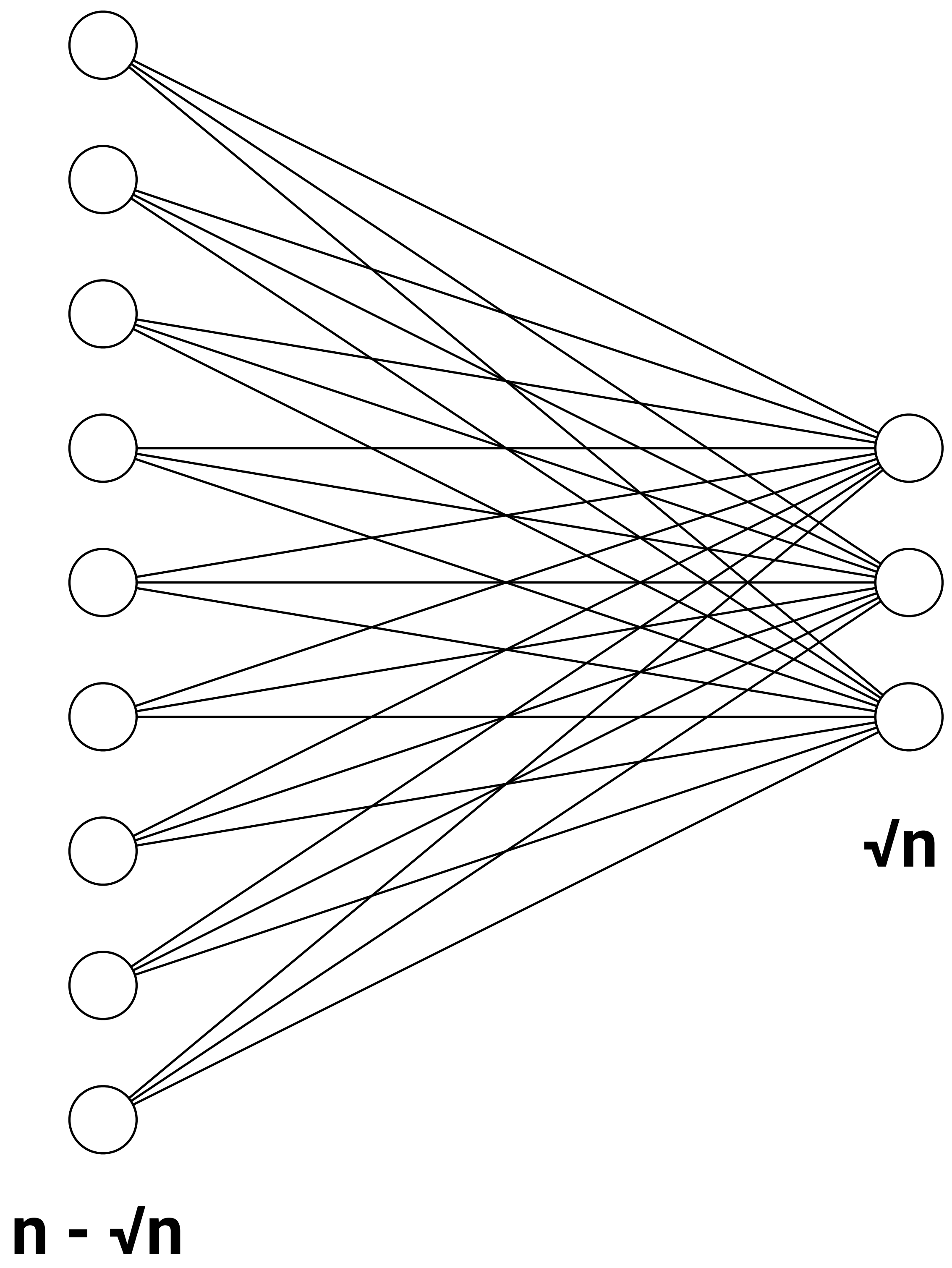
Randomized MIS: Luby's Algorithm

- Can we prove that a **constant fraction** of nodes gets **removed**?
  - **No**!



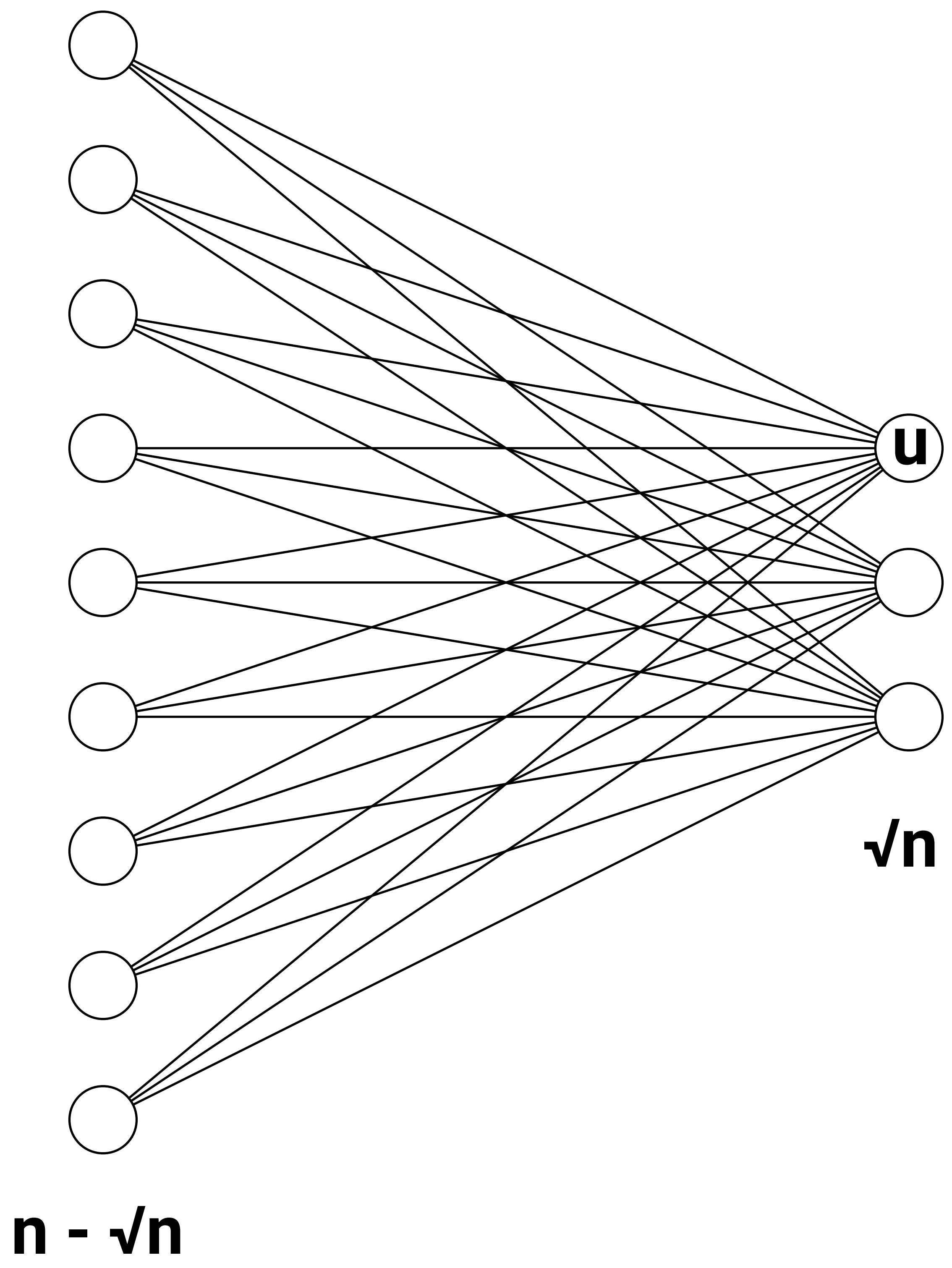
Randomized MIS: Luby's Algorithm

- Can we prove that a **constant fraction** of nodes gets **removed**?
  - **No**!



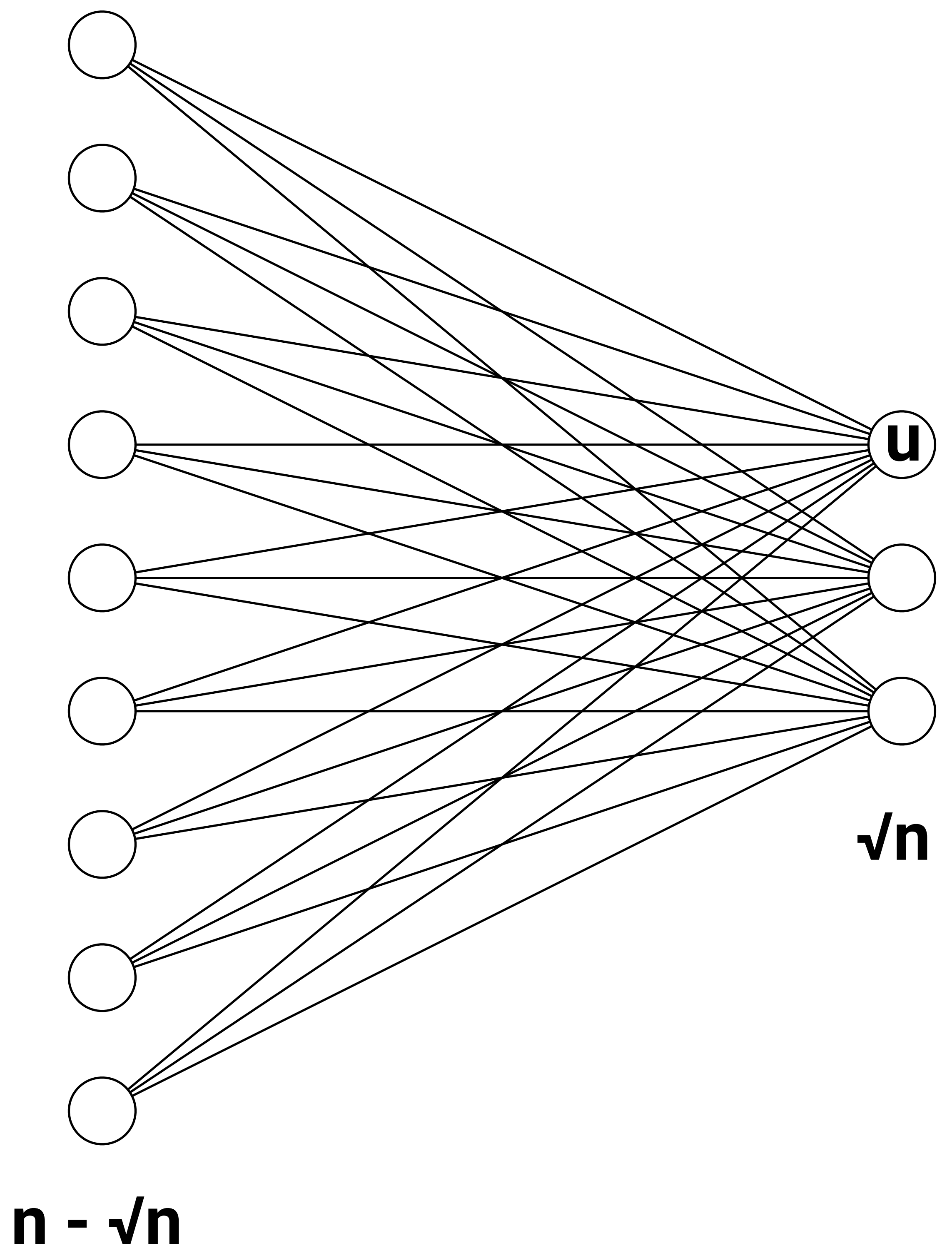
Randomized MIS: Luby's Algorithm

- Can we prove that a **constant fraction** of nodes gets **removed**?
  - **No**!



Randomized MIS: Luby's Algorithm

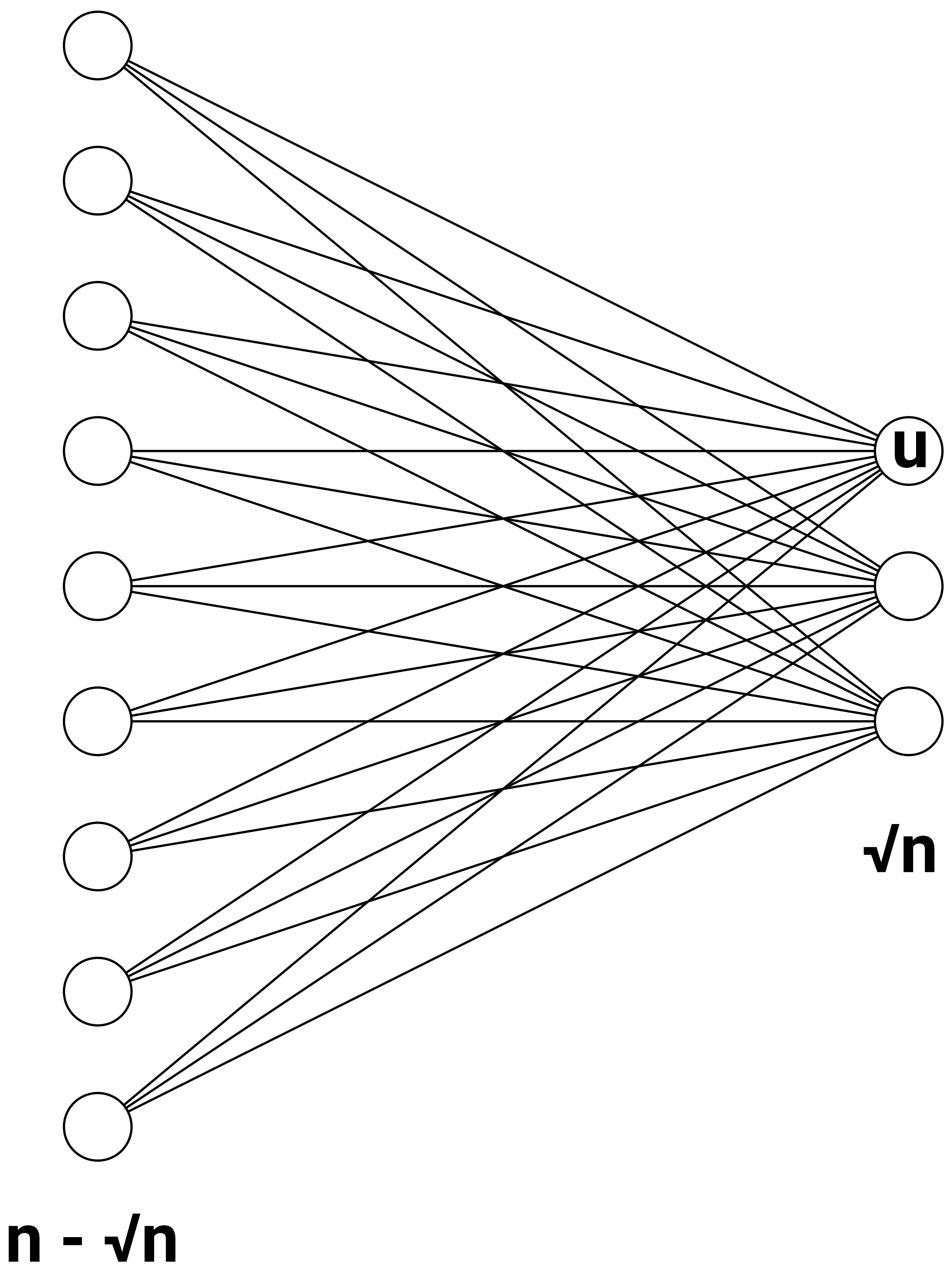
- Can we prove that a **constant fraction** of nodes gets **removed**?
  - **No**!



P(u joins MIS) ≈ 1 / n

Randomized MIS: Luby's Algorithm

- Can we prove that a **constant fraction** of nodes gets **removed**?
  - **No**!

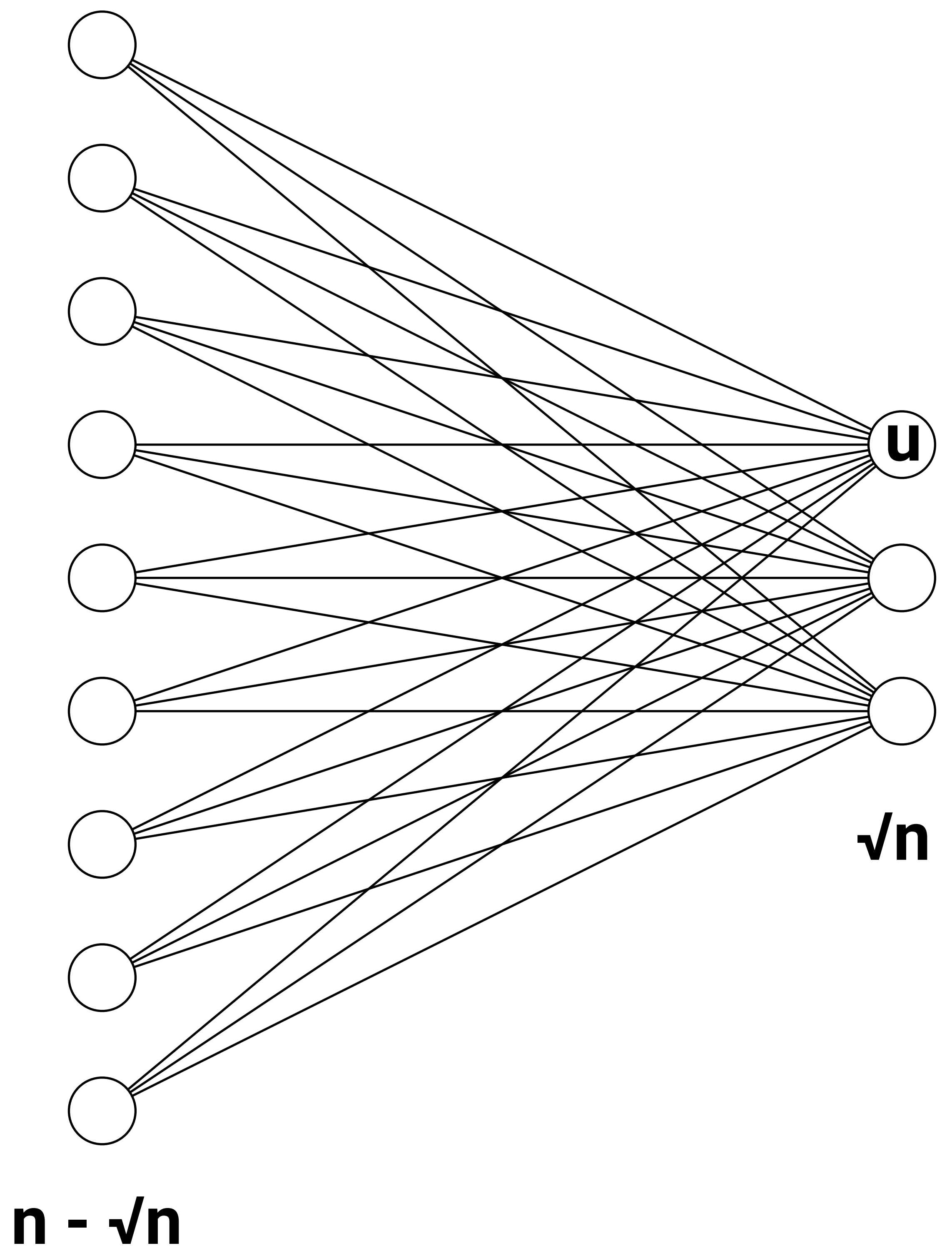


P(**u** joins **MIS** **)** ≈ **1 / n**

P(**some node of the right side** joins **MIS** **)** ≈ **1 / √n**

Randomized MIS: Luby's Algorithm

- Can we prove that a **constant fraction** of nodes gets **removed**?
  - **No**!



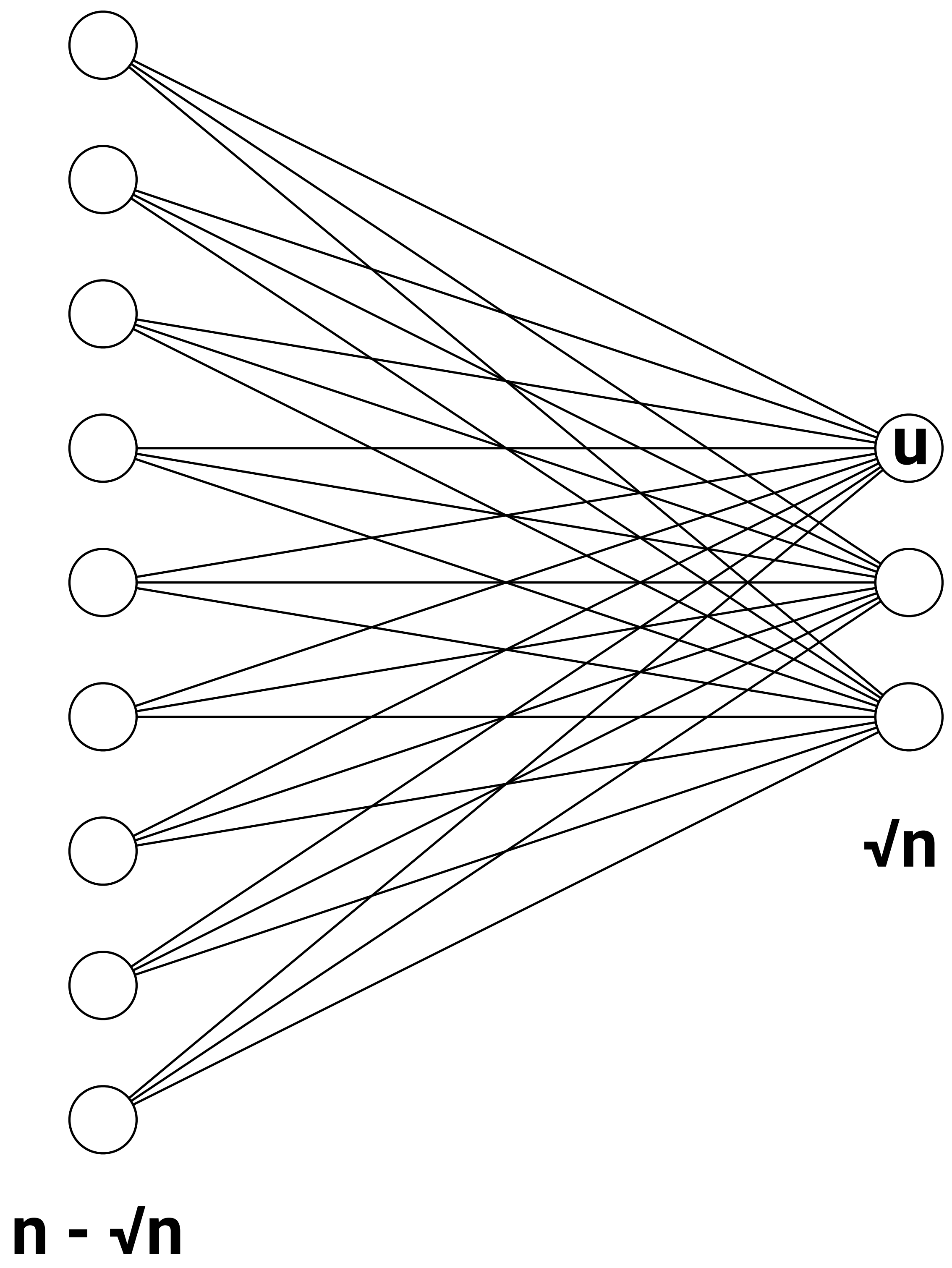
P(**u** joins **MIS**) ≈ **1 / n**

P(**some node of the right side** joins **MIS**) ≈ **1 / √n**

with high probability, **no node** of the **right** side joins the MIS

Randomized MIS: Luby's Algorithm

- Can we prove that a **constant fraction** of nodes gets **removed**?
  - **No**!



P(**u** joins **MIS** **)** ≈ **1 / n**

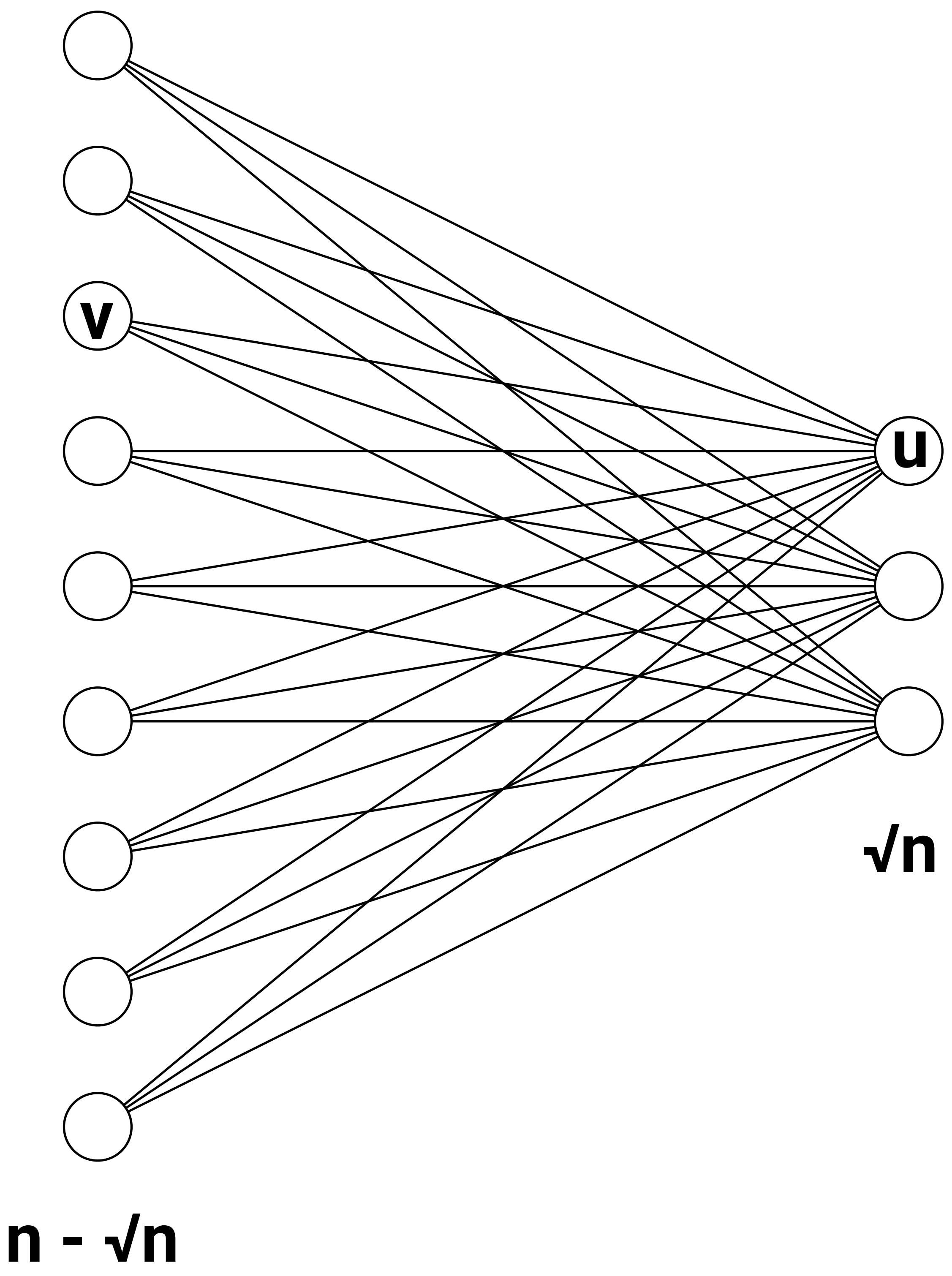
P(**some node of the right side** joins **MIS** **)** ≈ **1 / √n**

with high probability, **no node** of the **right** side joins the MIS

nodes of the left side only get removed if they join MIS

Randomized MIS: Luby's Algorithm

- Can we prove that a **constant fraction** of nodes gets **removed**?
  - **No**!



P(**u** joins **MIS** **)** ≈ **1 / n**

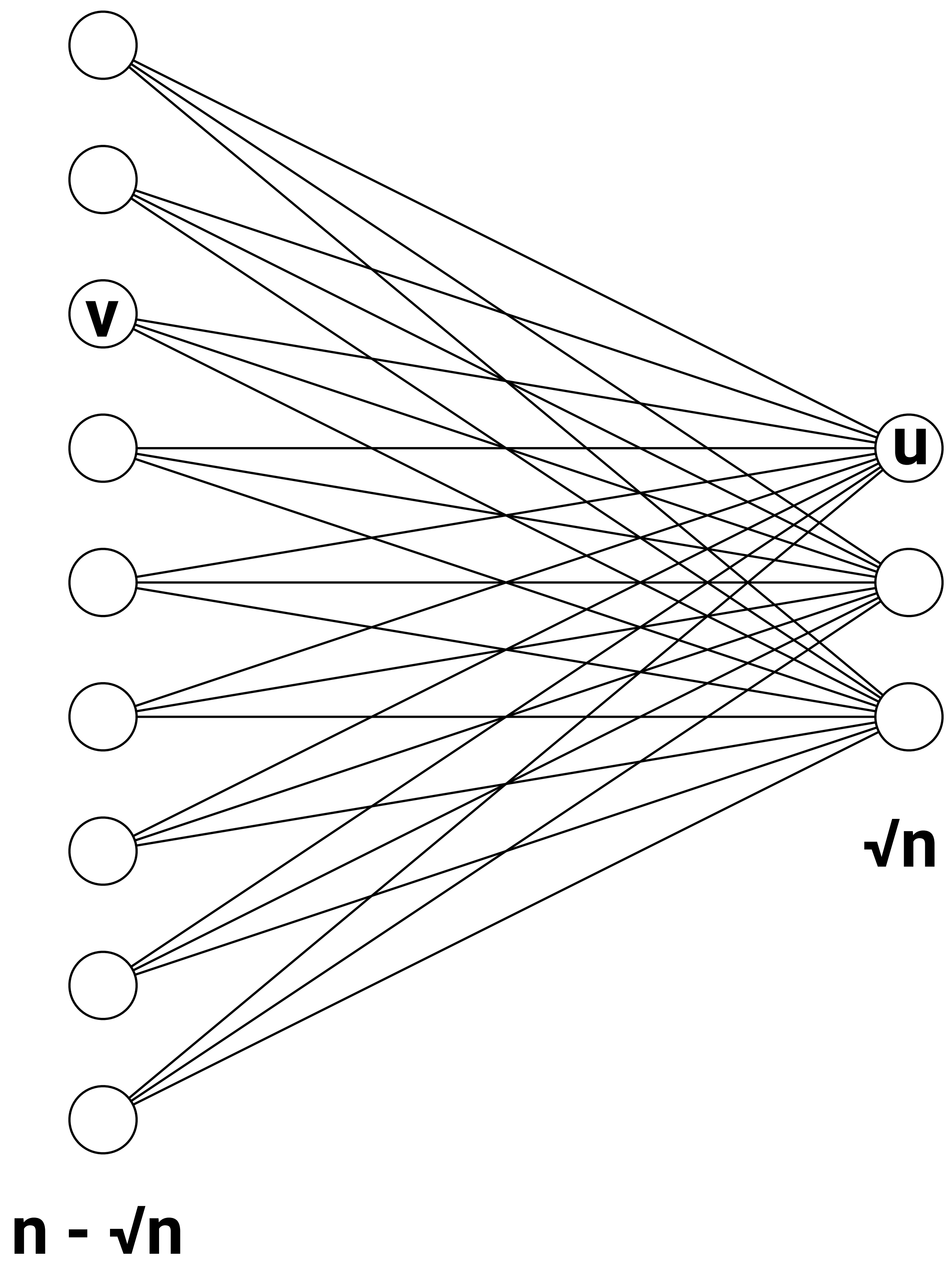
P(**some node of the right side** joins **MIS** **)** ≈ **1 / √n**

with high probability, **no node** of the **right** side joins the MIS

nodes of the left side only get removed if they join MIS

Randomized MIS: Luby's Algorithm

- Can we prove that a **constant fraction** of nodes gets **removed**?
  - **No**!



P(u joins MIS **)** ≈ **1 / n**

P(**some node of the right side** joins **MIS** **)** ≈ **1 / √n**

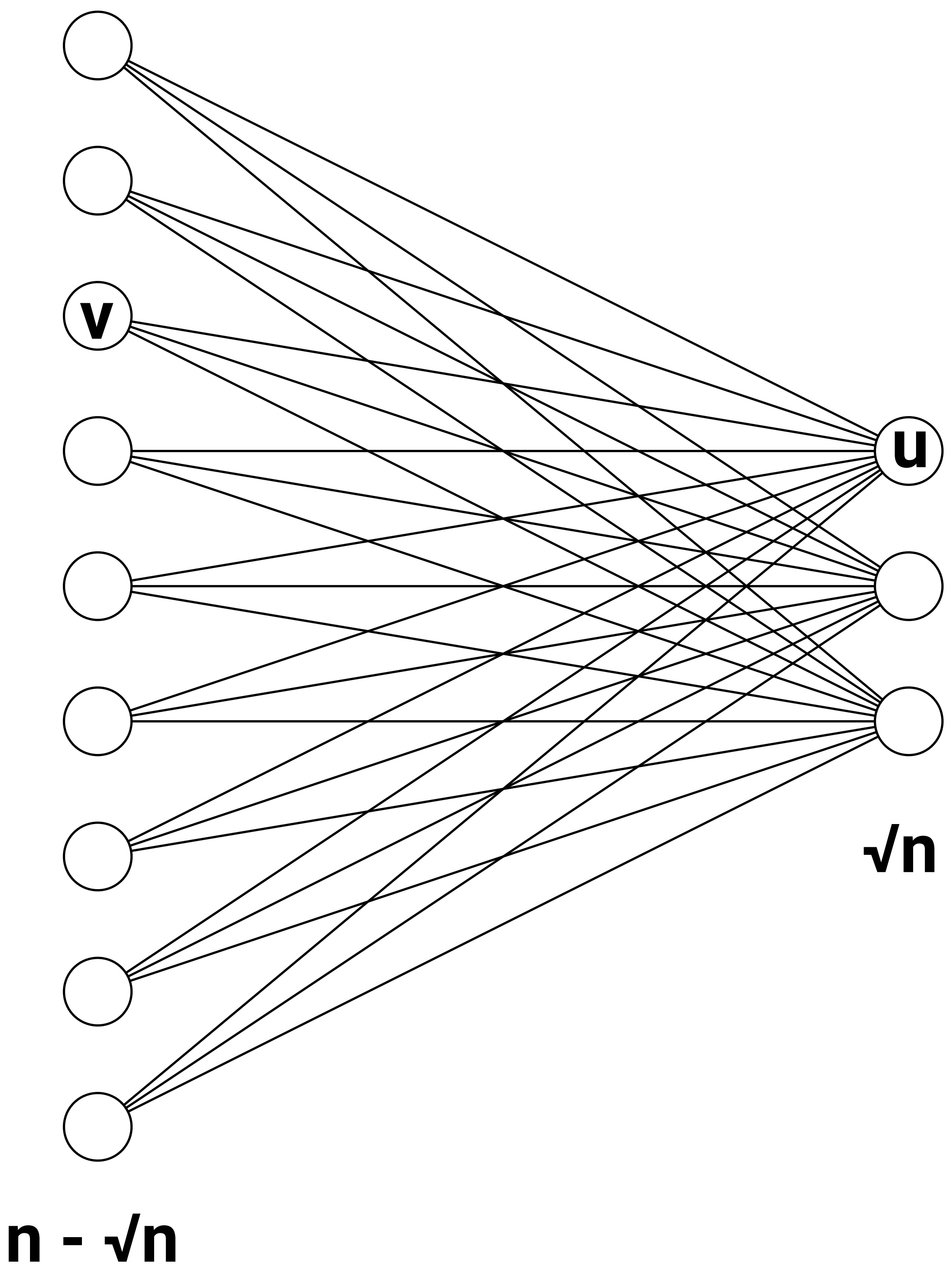
with high probability, **no node** of the **right** side joins the MIS

nodes of the left side only get removed if they join MIS

P(v joins **MIS** **)** ≈ **1 / √n**

Randomized MIS: Luby's Algorithm

- Can we prove that a **constant fraction** of nodes gets **removed**?
  - **No**!



P(**u** joins **MIS** **)** ≈ **1 / n**

P(**some node of the right side** joins **MIS** **)** ≈ **1 / √n**

with high probability, **no node** of the **right** side joins the MIS

nodes of the left side only get removed if they join MIS

P(**v** joins **MIS** **)** ≈ **1 / √n**

only a **1/√n fraction** of nodes join the MIS

Randomized MIS: Luby's Algorithm

Randomized MIS: Luby's Algorithm

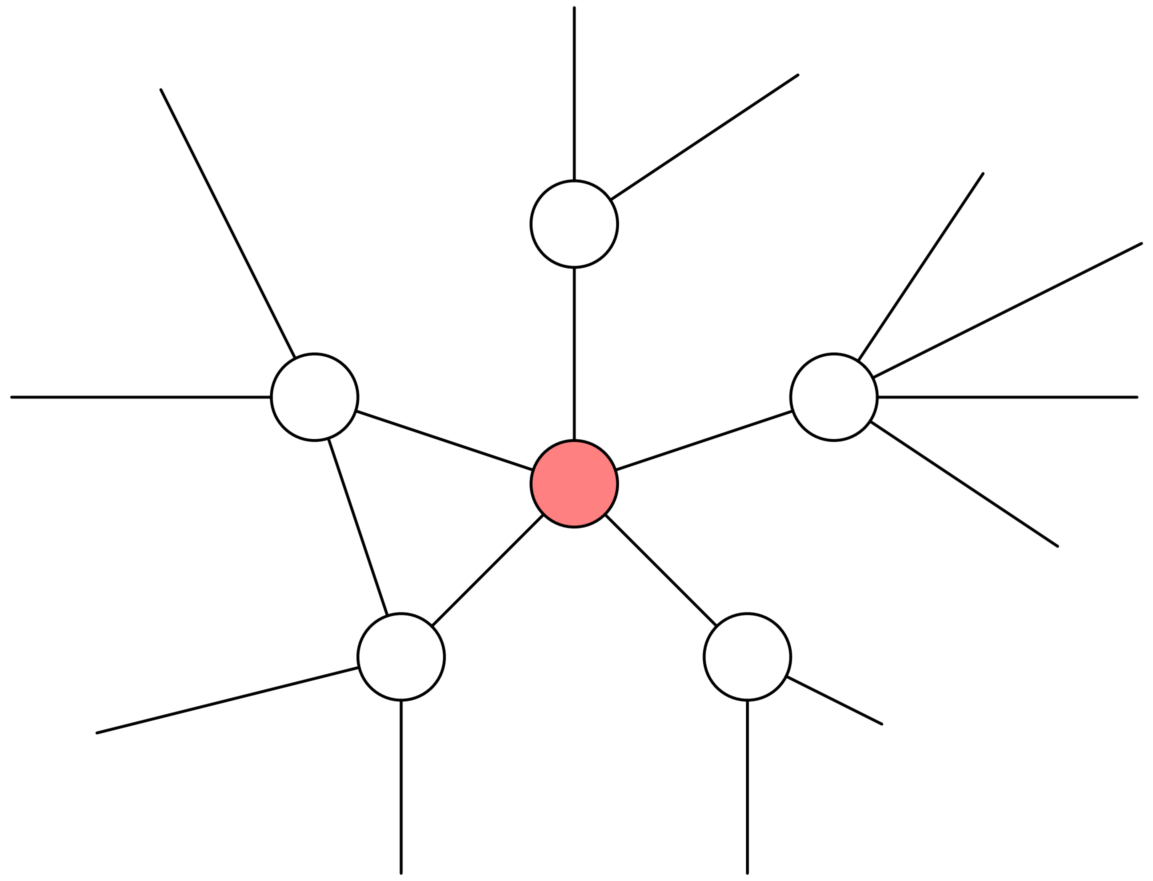
- **Lemma:** in **expectation**, at least **half** of the remaining **edges** are **removed**.

Randomized MIS: Luby's Algorithm

- **Lemma:** in **expectation**, at least **half** of the remaining **edges** are **removed**.
- An edge **{u, v}** gets removed if either **u or v** gets removed

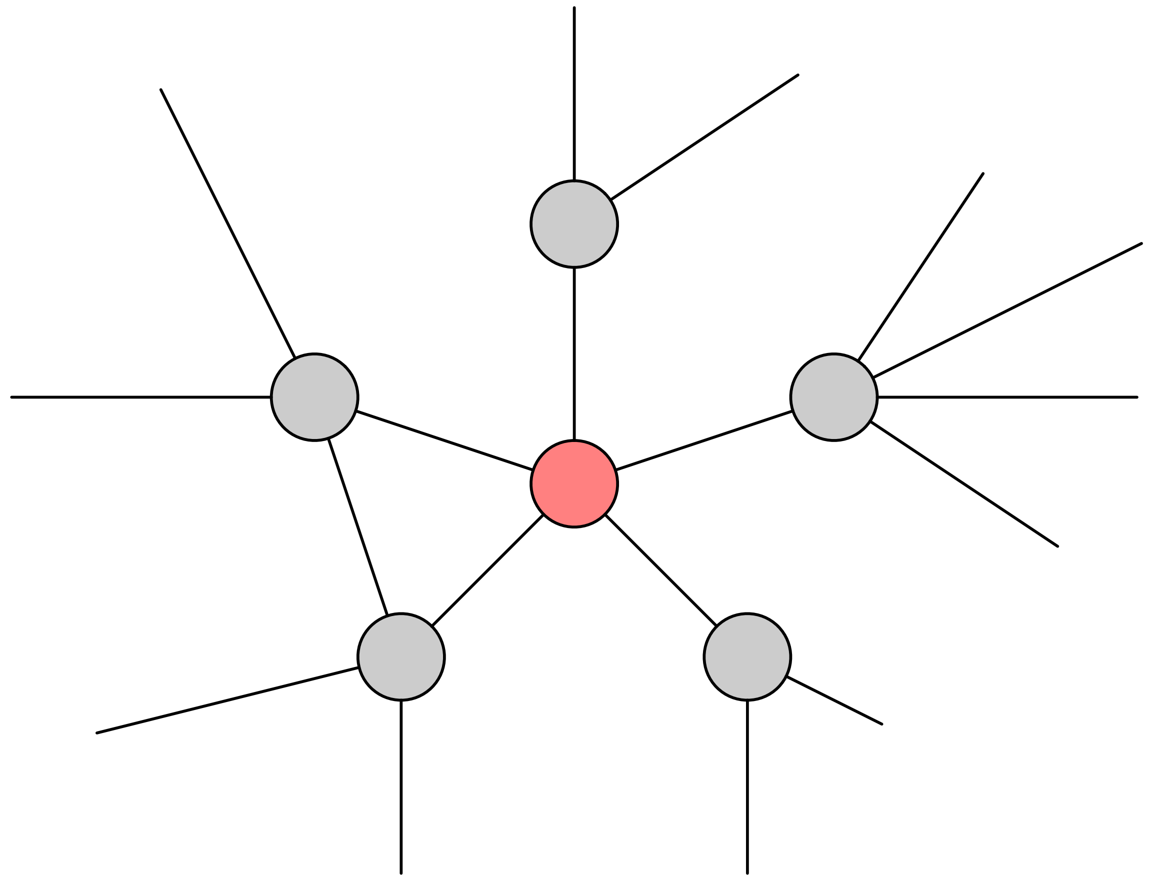
Randomized MIS: Luby's Algorithm

- **Lemma:** in **expectation**, at least **half** of the remaining **edges** are **removed**.
- An edge **{u, v}** gets removed if either **u or v** gets removed



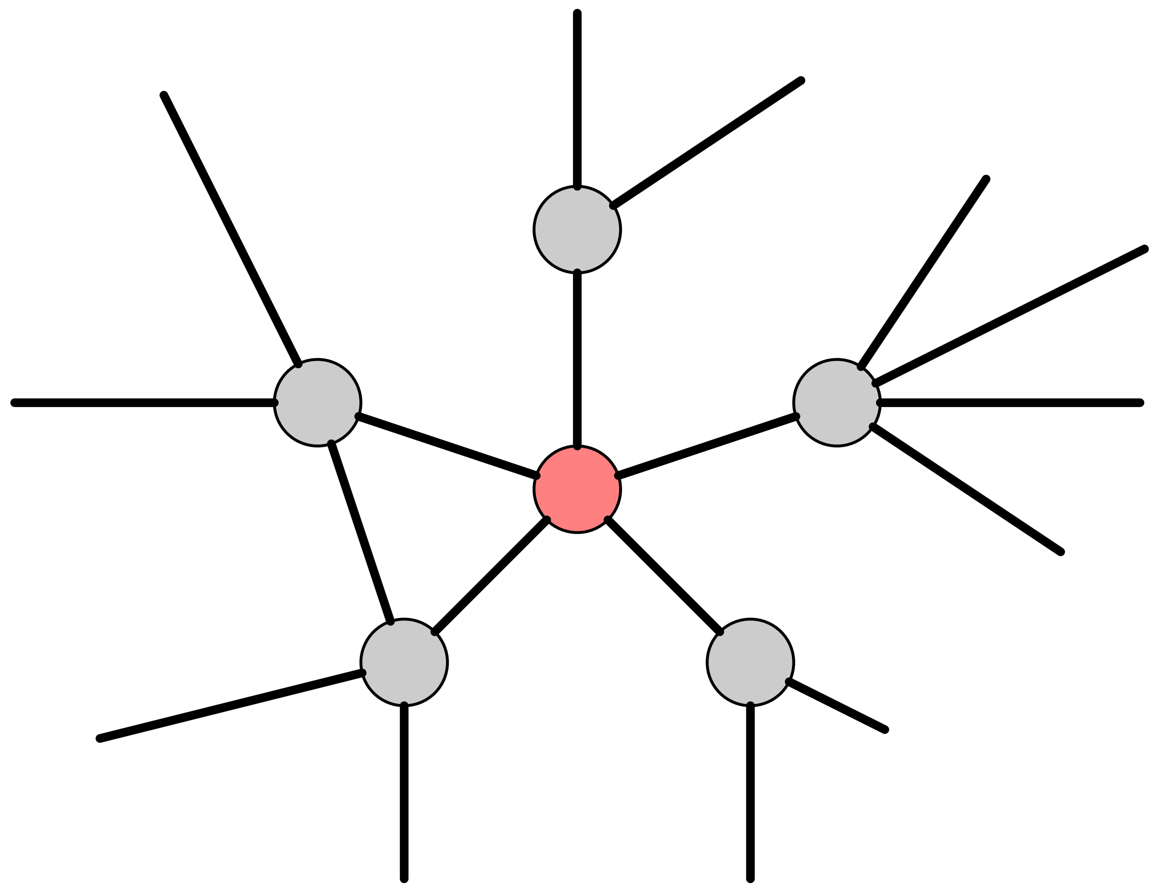
Randomized MIS: Luby's Algorithm

- **Lemma:** in **expectation**, at least **half** of the remaining **edges** are **removed**.
- An edge **{u, v}** gets removed if either **u or v** gets removed



Randomized MIS: Luby's Algorithm

- **Lemma:** in **expectation**, at least **half** of the remaining **edges** are **removed**.
- An edge **{u, v}** gets removed if either **u or v** gets removed



Randomized MIS: Luby's Algorithm

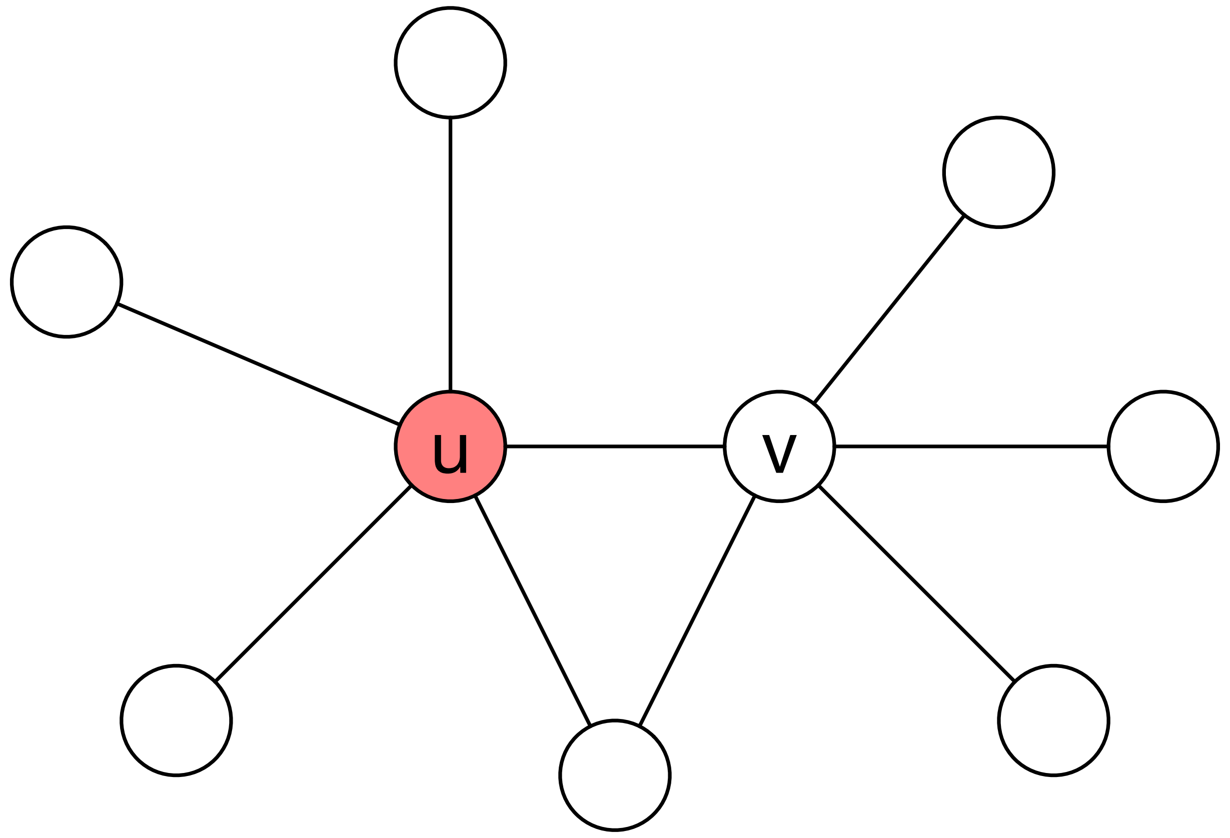
- **Lemma:** in **expectation**, at least **half** of the remaining **edges** are **removed**.
- An edge **{u, v}** gets removed if either **u or v** gets removed
- For each edge **{u, v}**, we define events $\mathcal{E}_{u,v}$ and $\mathcal{E}_{v,u}$

Randomized MIS: Luby's Algorithm

- **Lemma:** in **expectation**, at least **half** of the remaining **edges** are **removed**.
- An edge **{u, v}** gets removed if either **u** **or** **v** gets removed
- For each edge **{u, v}**, we define events $\mathcal{E}_{u,v}$ and $\mathcal{E}_{v,u}$
  - $\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$

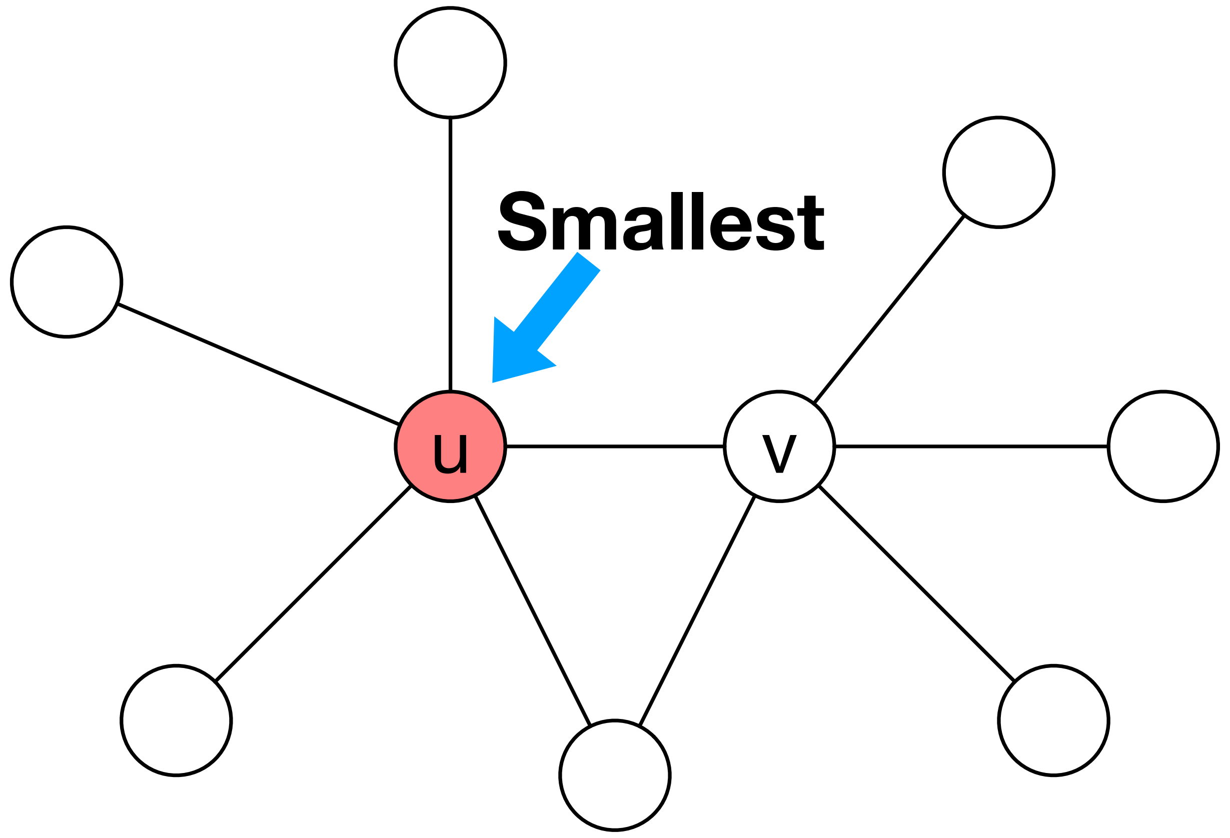
Randomized MIS: Luby's Algorithm

- **Lemma:** in **expectation**, at least **half** of the remaining **edges** are **removed**.
- An edge **{u, v}** gets removed if either **u or v** gets removed
- For each edge **{u, v}**, we define events $\mathcal{E}_{u,v}$ and $\mathcal{E}_{v,u}$
  - $\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$



Randomized MIS: Luby's Algorithm

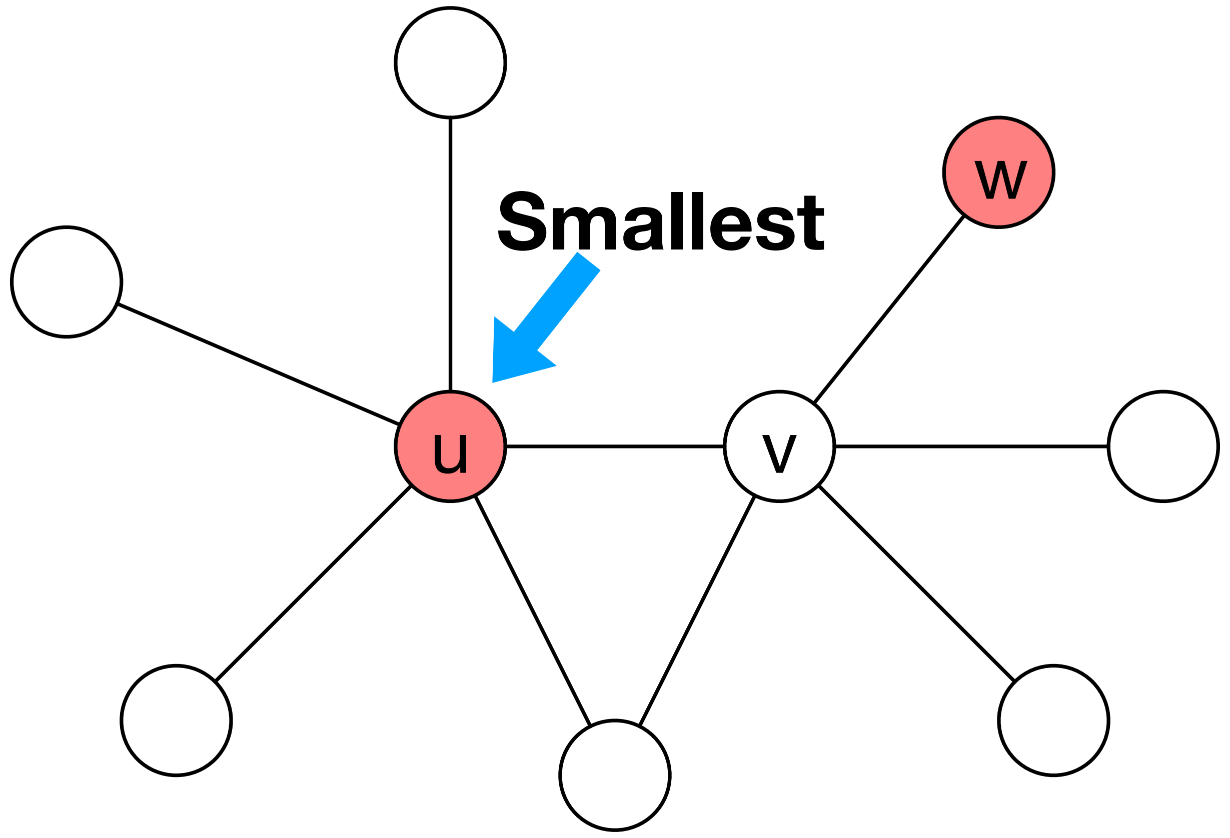
- **Lemma:** in **expectation**, at least **half** of the remaining **edges** are **removed**.
- An edge **{u, v}** gets removed if either **u or v** gets removed
- For each edge **{u, v}**, we define events $\mathcal{E}_{u,v}$ and $\mathcal{E}_{v,u}$
  - $\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$



$\mathcal{E}_{u,v}$ is **true**

Randomized MIS: Luby's Algorithm

- **Lemma:** in **expectation**, at least **half** of the remaining **edges** are **removed**.
- An edge **{u, v}** gets removed if either **u or v** gets removed
- For each edge **{u, v}**, we define events $\mathcal{E}_{u,v}$ and $\mathcal{E}_{v,u}$
  - $\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$

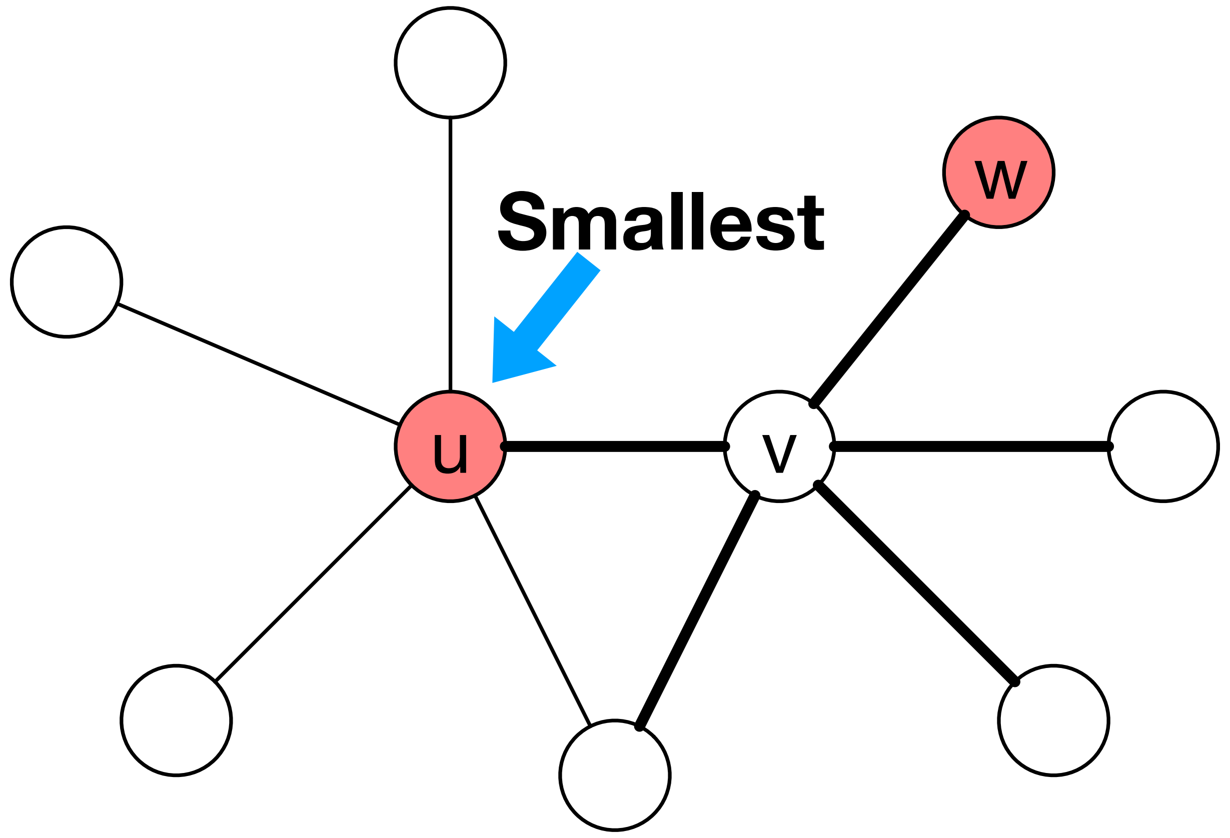


$\mathcal{E}_{u,v}$ is **true**

$\mathcal{E}_{w,v}$ is **false**

Randomized MIS: Luby's Algorithm

- **Lemma:** in **expectation**, at least **half** of the remaining **edges** are **removed**.
- An edge **{u, v}** gets removed if either **u or v** gets removed
- For each edge **{u, v}**, we define events $\mathcal{E}_{u,v}$ and $\mathcal{E}_{v,u}$
  - $\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$



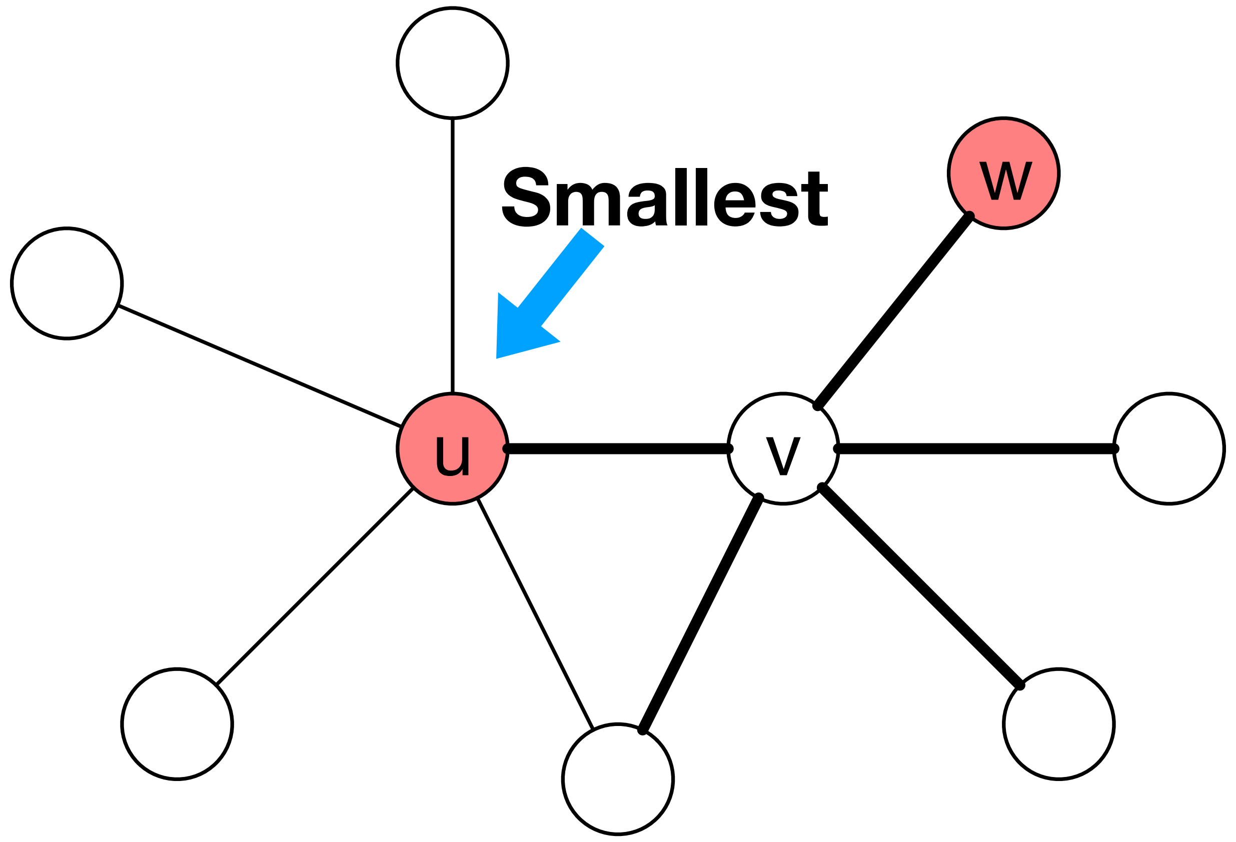
$\mathcal{E}_{u,v}$ is **true** ⇒ all edges of **v** get removed "**because of u**"

$\mathcal{E}_{w,v}$ is **false**

Randomized MIS: Luby's Algorithm

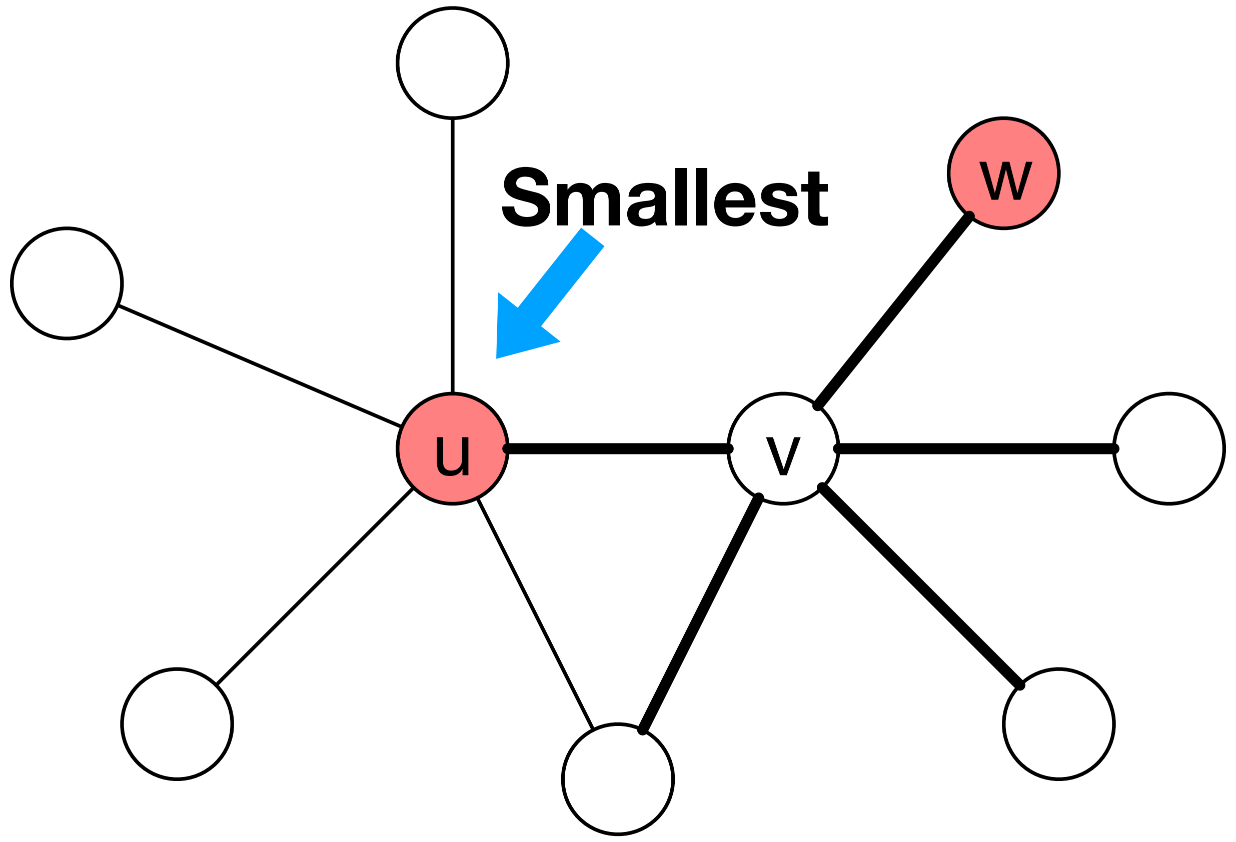
Randomized MIS: Luby's Algorithm

- $\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$



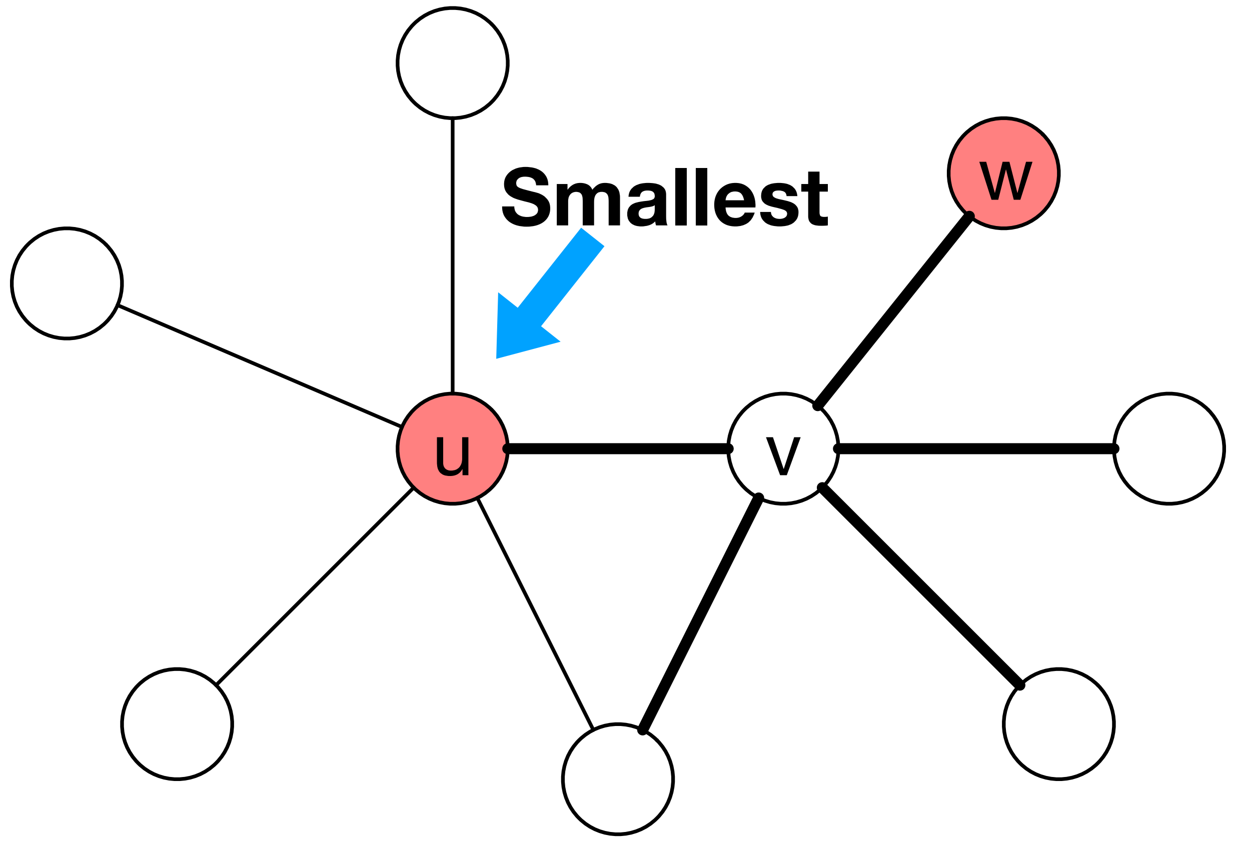
Randomized MIS: Luby's Algorithm

- $\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$
- $X_{u,v}$:= deg(v) if $\mathcal{E}_{u,v}$ holds, 0 otherwise



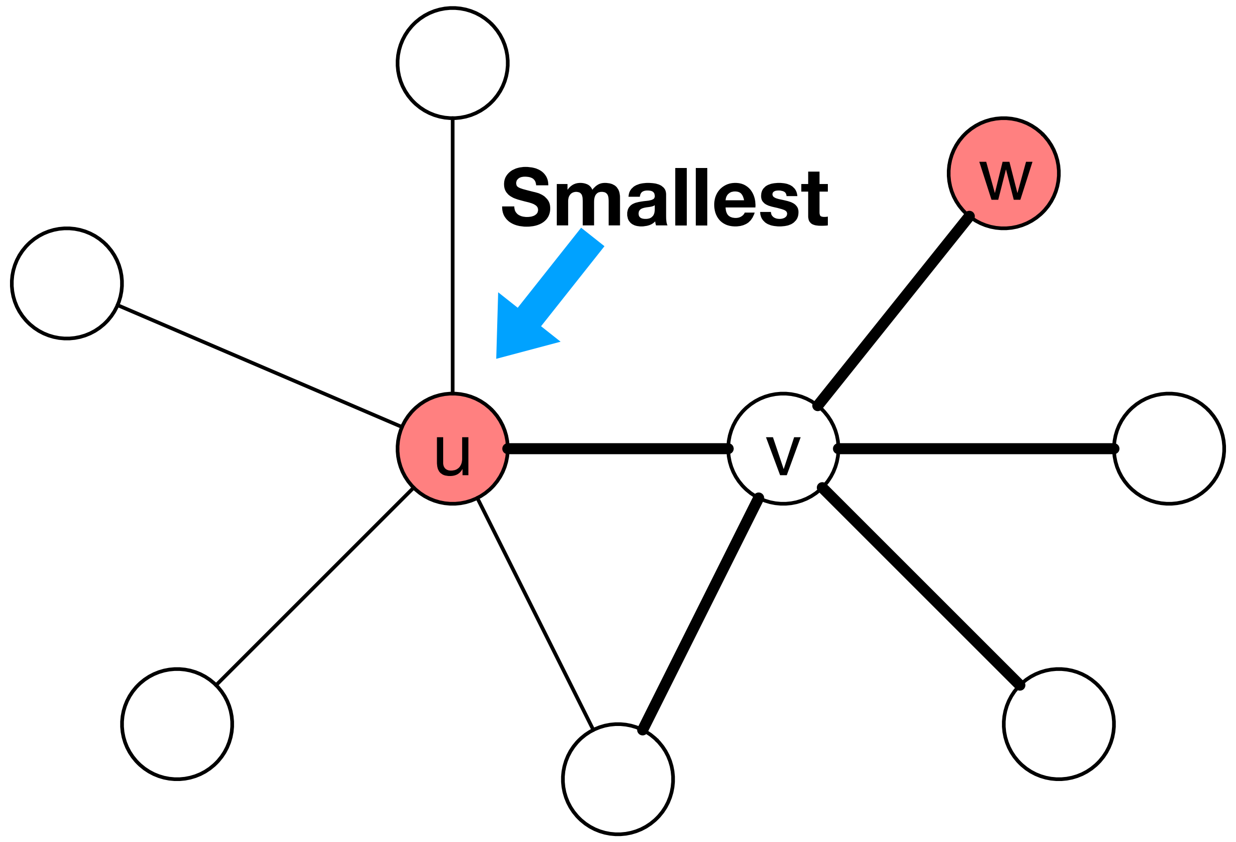
Randomized MIS: Luby's Algorithm

- $\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$
- $X_{u,v} :=$ deg(v) if $\mathcal{E}_{u,v}$ holds, 0 otherwise
- $X := \sum_{\{u, v\} \in E} (X_{u,v} + X_{v,u})$



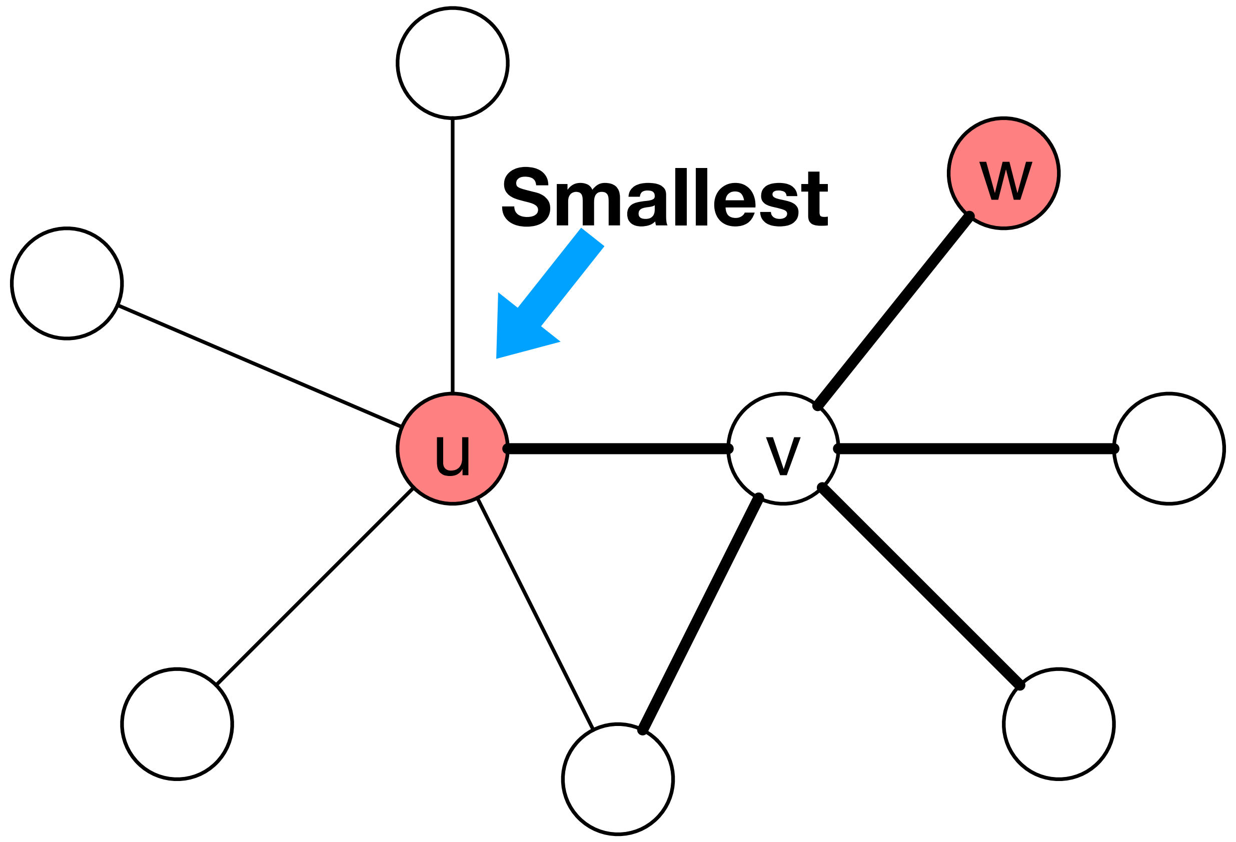
Randomized MIS: Luby's Algorithm

- $\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$
- $X_{u,v} := \deg(v)$ if $\mathcal{E}_{u,v}$ holds, 0 otherwise
- $X := \sum_{\{u, v\} \in E} (X_{u,v} + X_{v,u})$
- **Claim:** $X \leq 2 \cdot$ **# deleted edges**



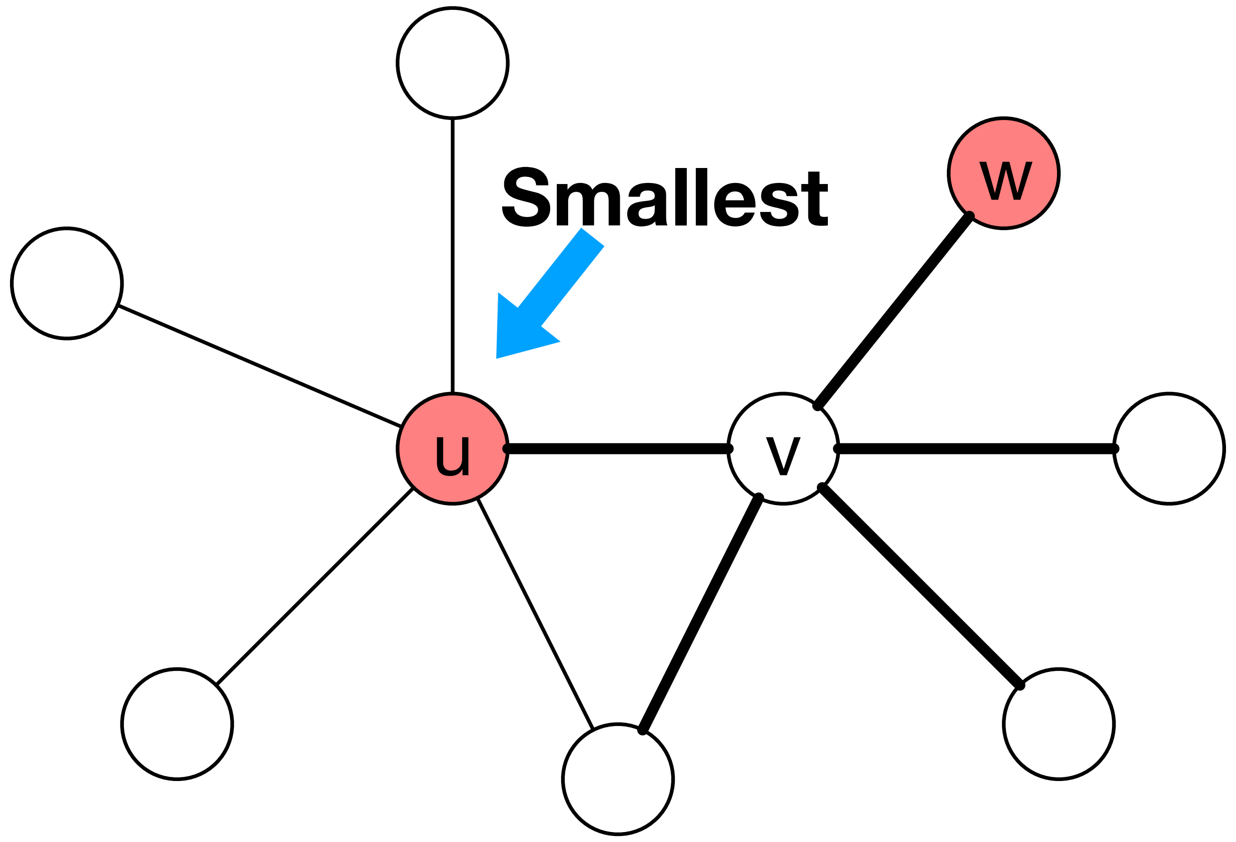
Randomized MIS: Luby's Algorithm

- $\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$
- $X_{u,v} := \deg(v)$ if $\mathcal{E}_{u,v}$ holds, 0 otherwise
- $X := \sum_{\{u,v\} \in E} (X_{u,v} + X_{v,u})$
- **Claim:** $X \leq 2 \cdot$ **# deleted edges**
    - For each node **v**, at most one incident edges satisfies $\mathcal{E}_{u,v}$



Randomized MIS: Luby's Algorithm

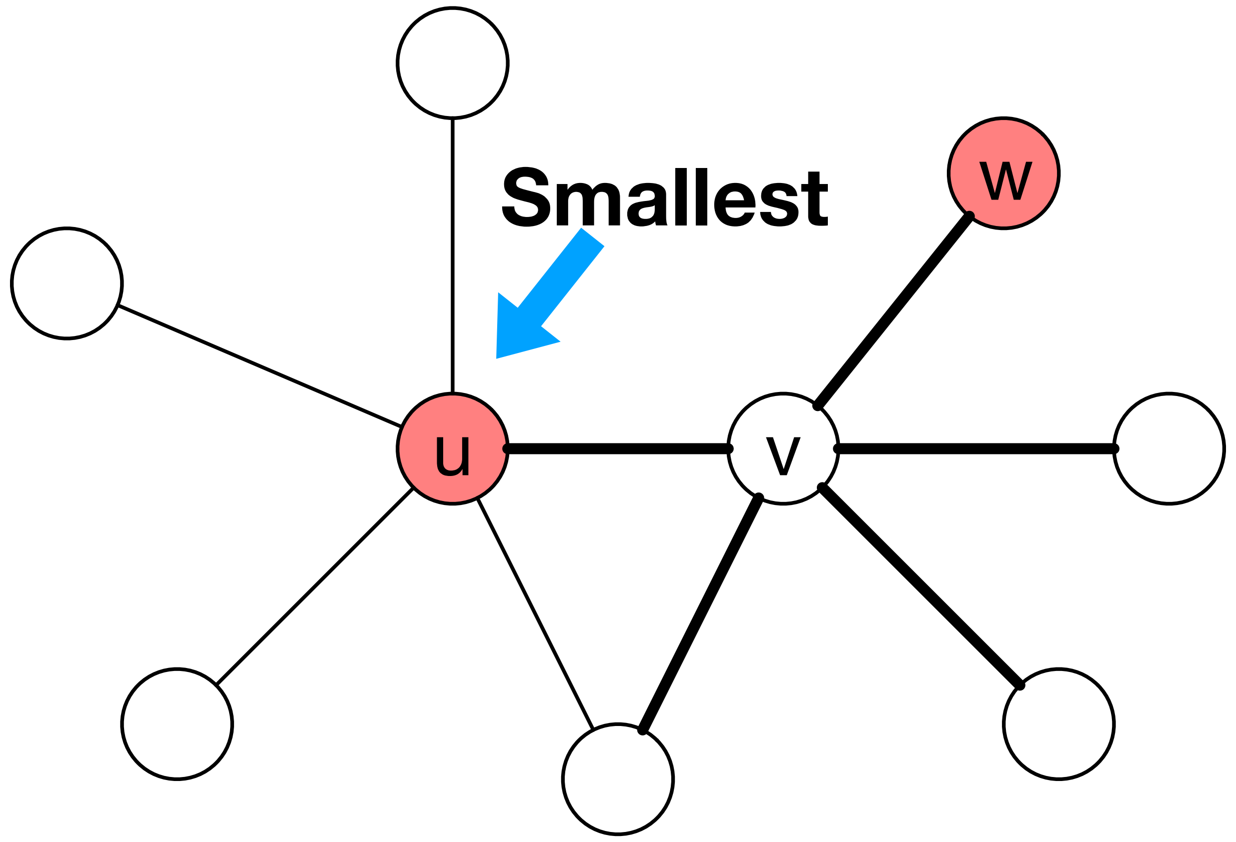
- $\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$
- $X_{u,v} := \deg(v)$ if $\mathcal{E}_{u,v}$ holds, 0 otherwise
- $X := \sum_{\{u, v\} \in E} (X_{u,v} + X_{v,u})$
- **Claim:** $X \leq 2 \cdot$ # deleted edges
  - For each node v, at most one incident edges satisfies $\mathcal{E}_{u,v}$



$\mathcal{E}_{u,v}$ is **true**

Randomized MIS: Luby's Algorithm

- $\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$
- $X_{u,v}$:= deg(v) if $\mathcal{E}_{u,v}$ holds, 0 otherwise
- $$X := \sum_{\{u, v\} \in E} (X_{u,v} + X_{v,u})$$
- **Claim:** $X \leq 2 \cdot$ **# deleted edges**
    - For each node v, at most one incident edges satisfies $\mathcal{E}_{u,v}$

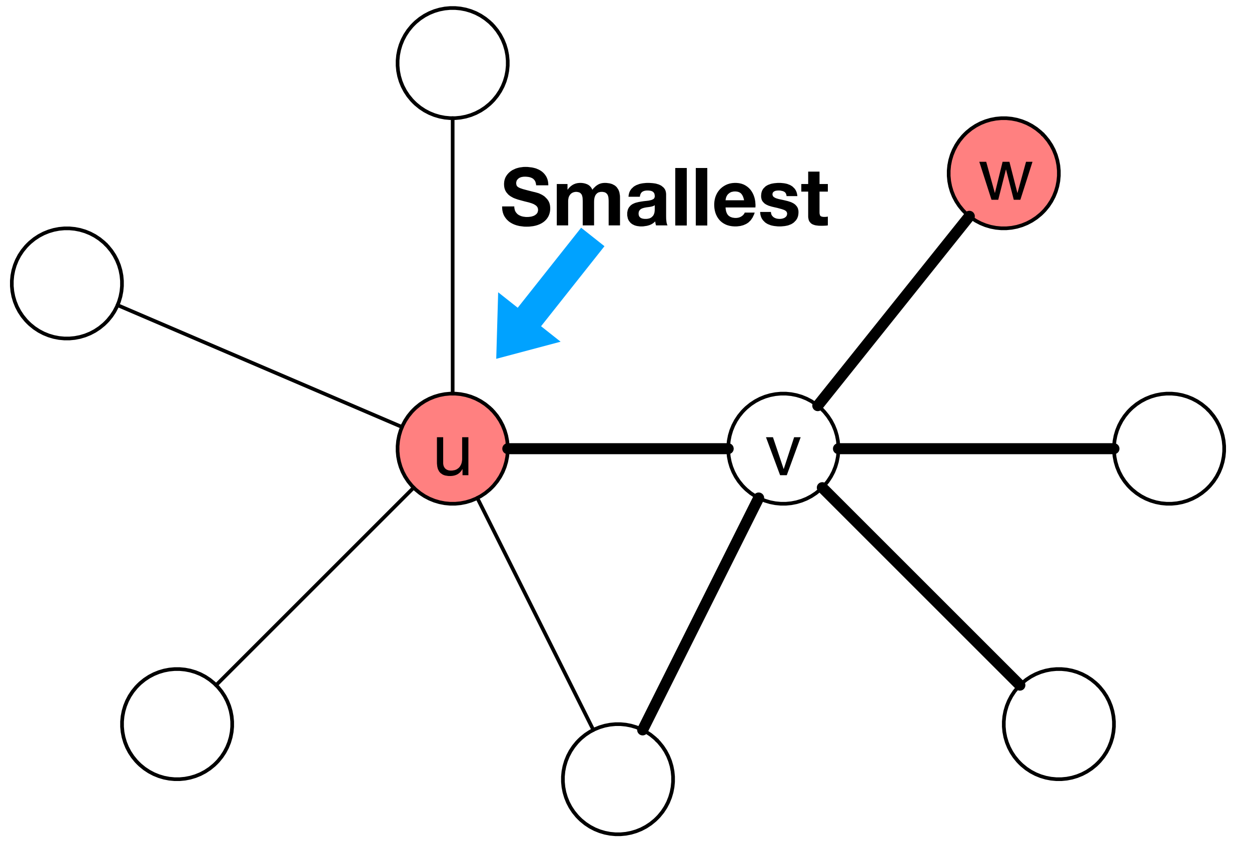


$\mathcal{E}_{u,v}$ is **true**

$X_{u,v}$ is **deg(v)**

Randomized MIS: Luby's Algorithm

- $\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$
- $X_{u,v}$:= deg(v) if $\mathcal{E}_{u,v}$ holds, 0 otherwise
- $X := \sum_{\{u, v\} \in E} (X_{u,v} + X_{v,u})$
- **Claim:** $X \leq 2 \cdot$ **# deleted edges**
  - For each node v, at most one incident edges satisfies $\mathcal{E}_{u,v}$



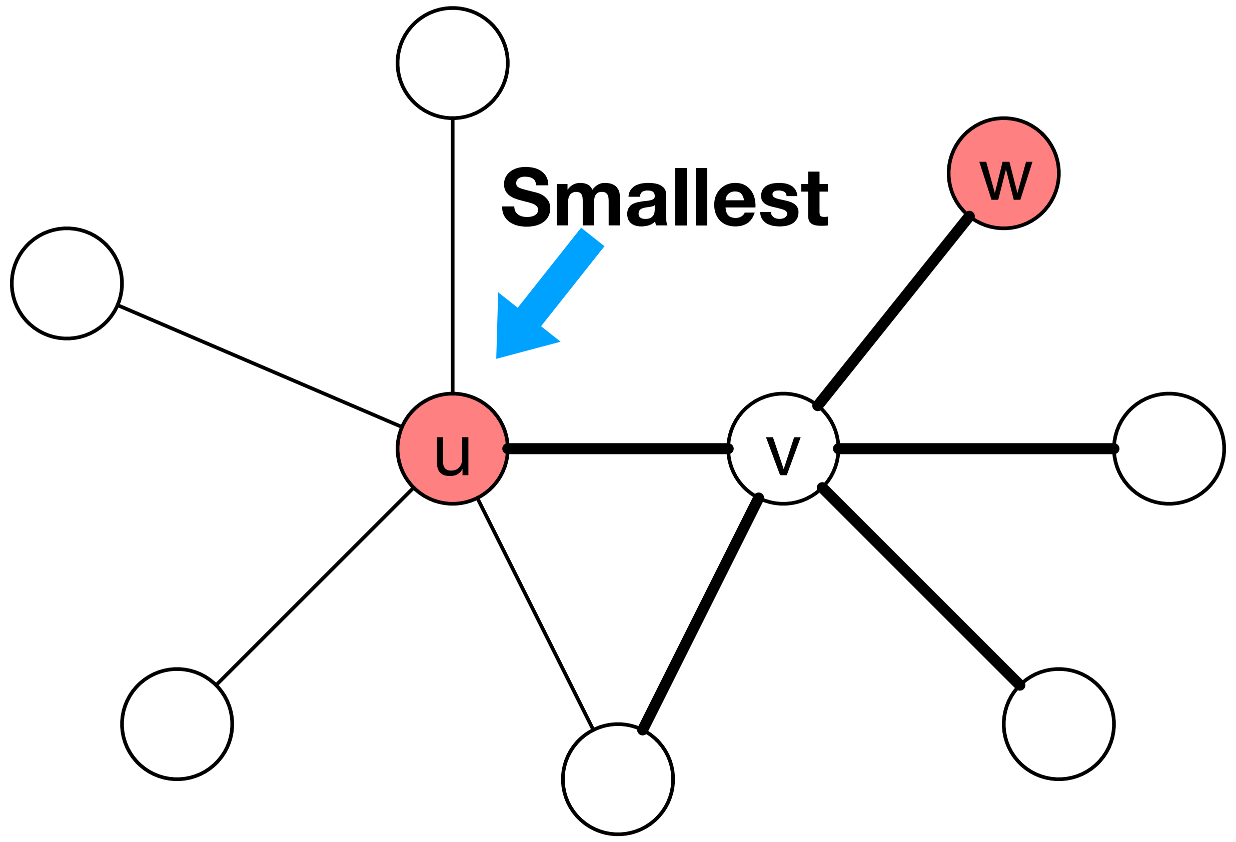
$\mathcal{E}_{u,v}$ is **true**

$X_{u,v}$ is **deg(v)**

$\mathcal{E}_{w,v}$ is **false**

Randomized MIS: Luby's Algorithm

- $\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$
- $X_{u,v} := \deg(v)$ if $\mathcal{E}_{u,v}$ holds, 0 otherwise
- $X := \sum_{\{u, v\} \in E} (X_{u,v} + X_{v,u})$
- **Claim:** $X \leq 2 \cdot$ **# deleted edges**
  - For each node v, at most one incident edges satisfies $\mathcal{E}_{u,v}$



$\mathcal{E}_{u,v}$ is **true**

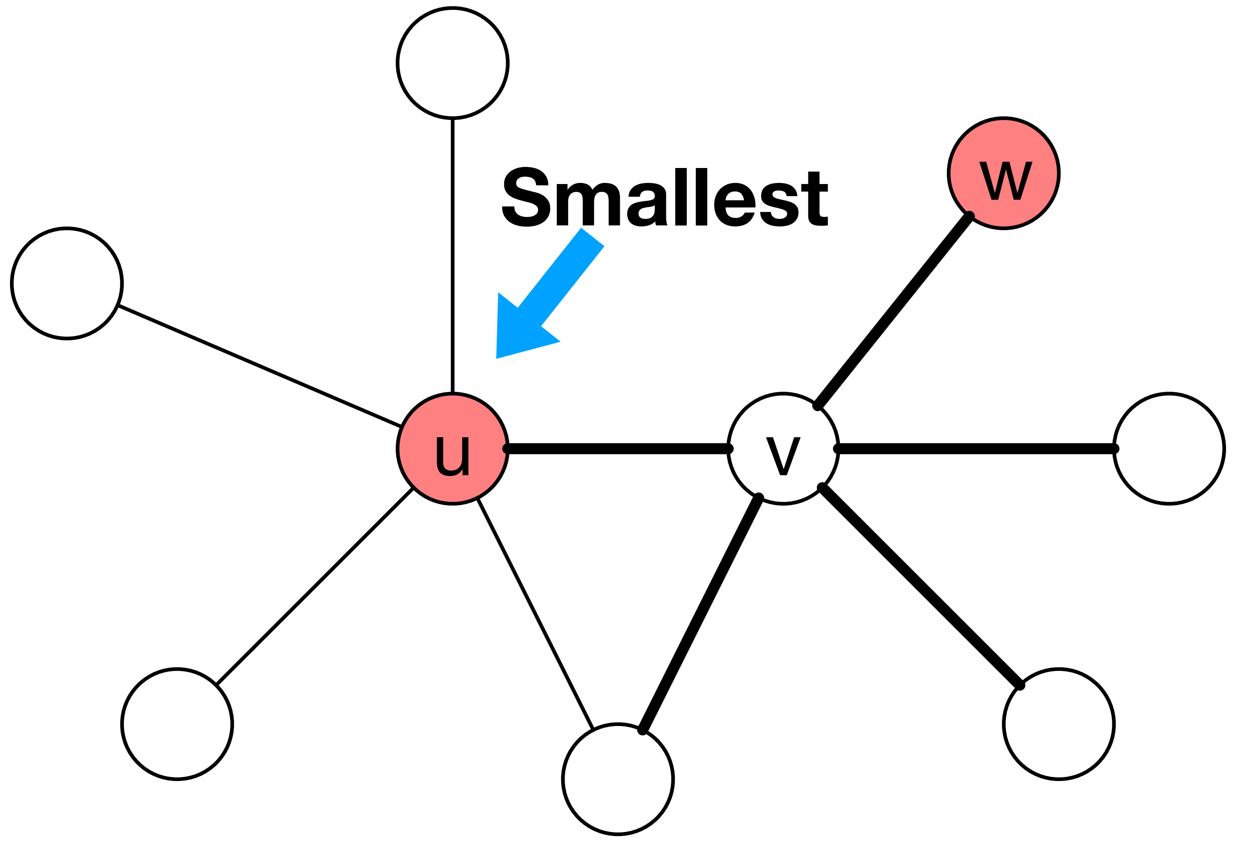
$X_{u,v}$ is **deg(v)**

$\mathcal{E}_{w,v}$ is **false**

$X_{w,v}$ is **0**

Randomized MIS: Luby's Algorithm

- $\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$
- $X_{u,v}$:= deg(v) if $\mathcal{E}_{u,v}$ holds, 0 otherwise
- $X := \sum_{\{u, v\} \in E} (X_{u,v} + X_{v,u})$
- **Claim:** $X \leq 2 \cdot$ **# deleted edges**
  - For each node v, at most one incident edges satisfies $\mathcal{E}_{u,v}$
  - so every edge incident to v is **counted once**, **for v**



$\mathcal{E}_{u,v}$ is **true**

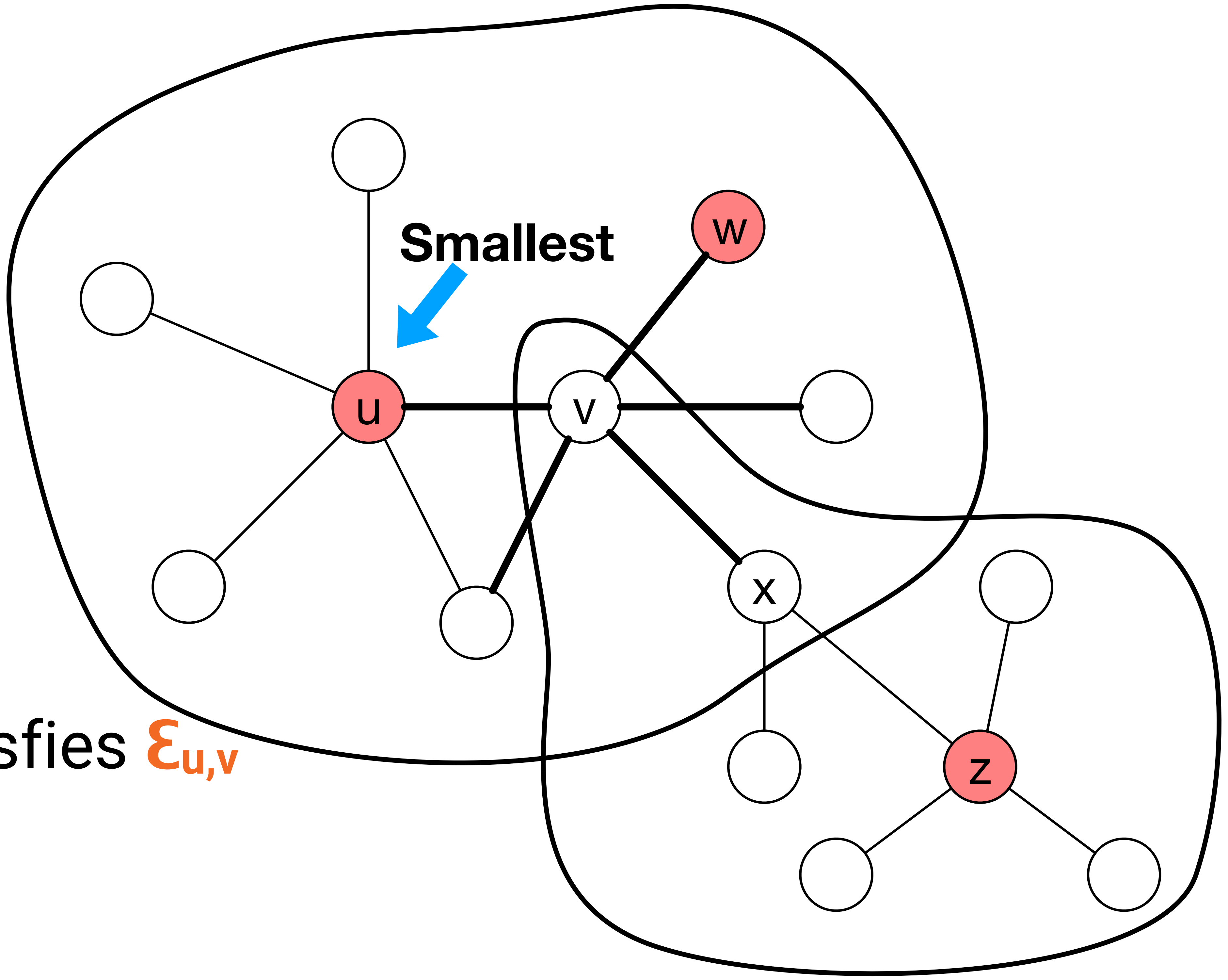
$X_{u,v}$ is **deg(v)**

$\mathcal{E}_{w,v}$ is **false**

$X_{w,v}$ is **0**

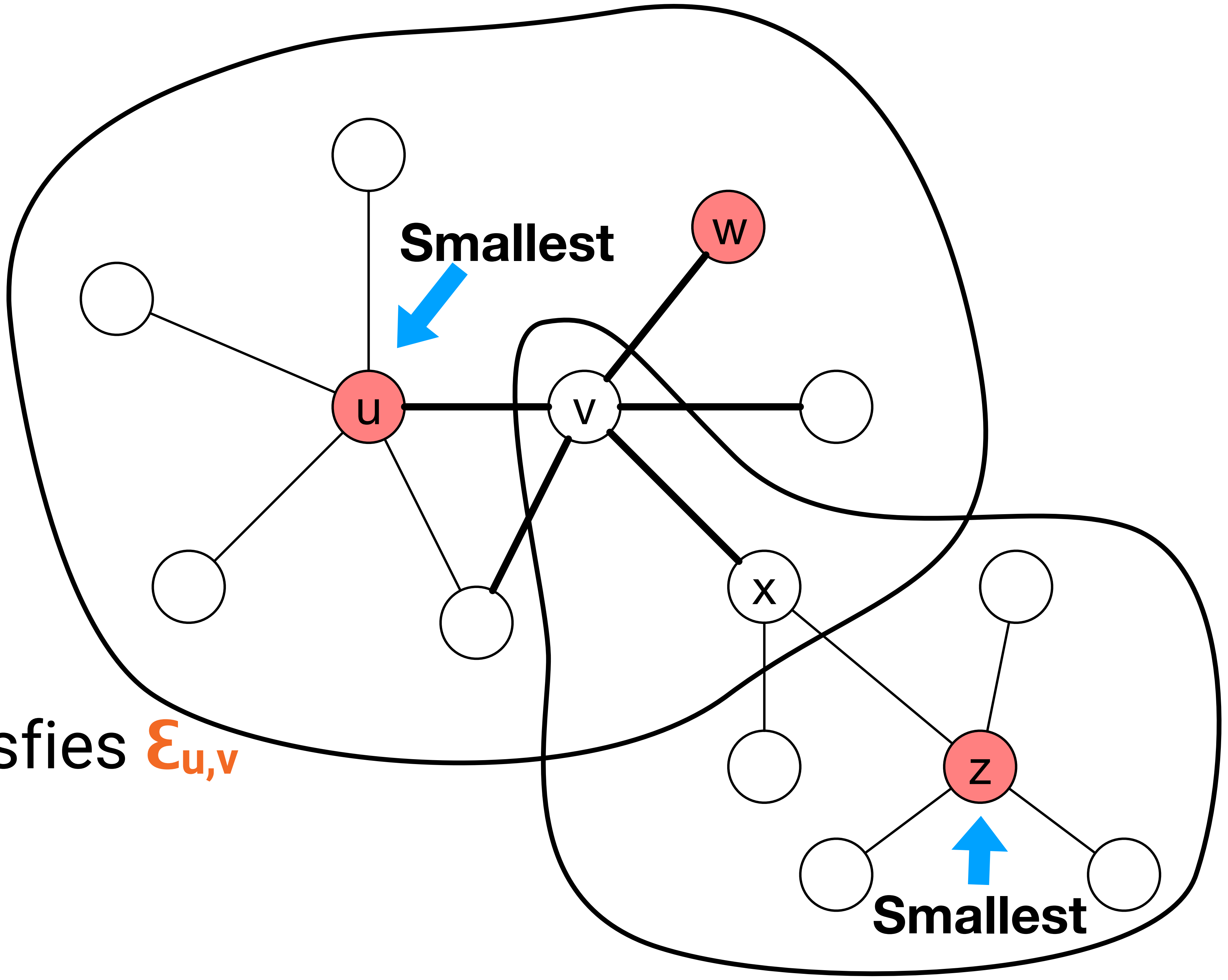
Randomized MIS: Luby's Algorithm

- $\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$
- $X_{u,v} := \deg(v)$ if $\mathcal{E}_{u,v}$ holds, 0 otherwise
- $X := \sum_{\{u, v\} \in E} (X_{u,v} + X_{v,u})$
- **Claim:** $X \leq 2 \cdot$ **# deleted edges**
  - For each node v, at most one incident edges satisfies $\mathcal{E}_{u,v}$
  - so every edge incident to v is **counted once**, **for v**



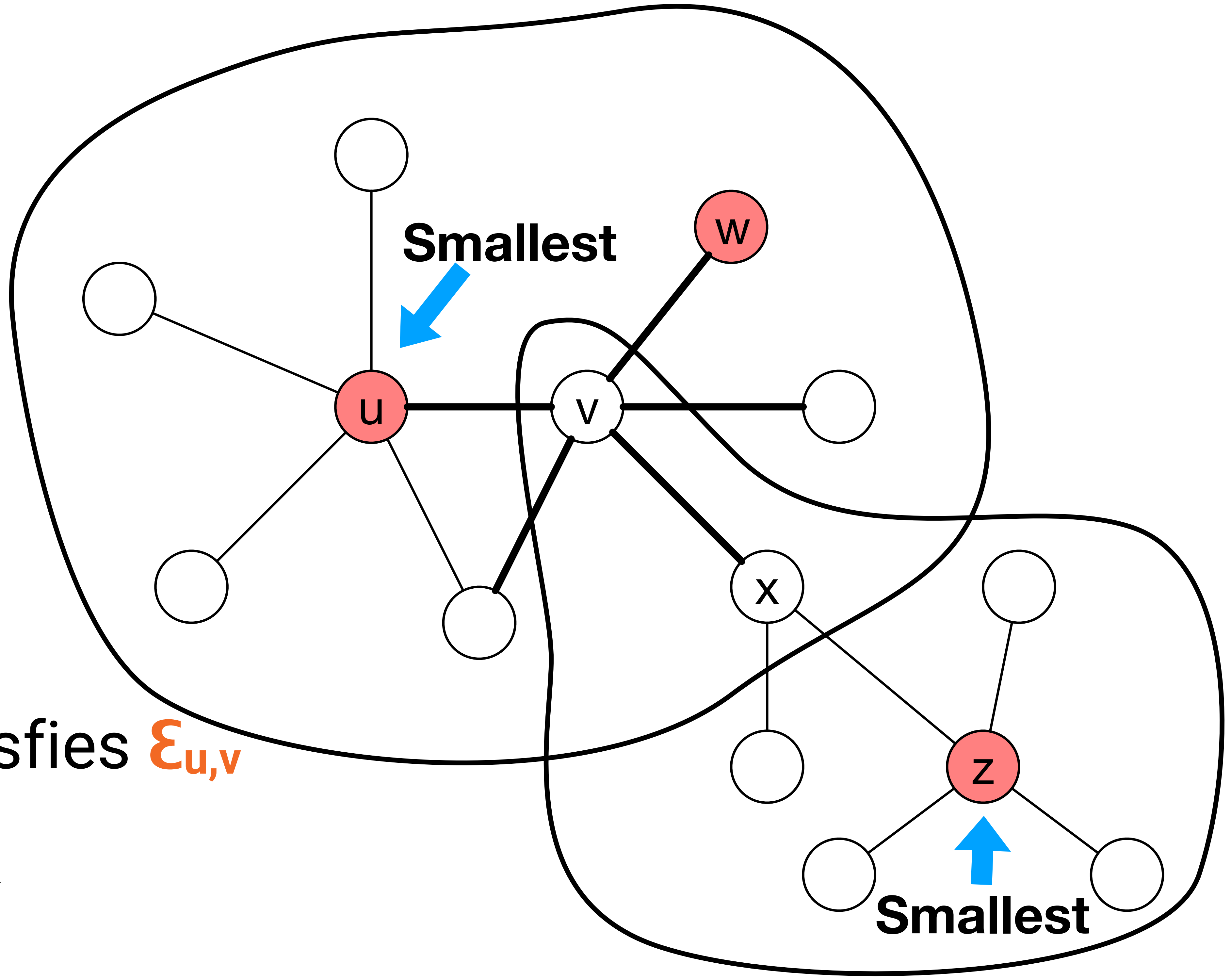
Randomized MIS: Luby's Algorithm

- $\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$
- $X_{u,v} :=$ deg(v) if $\mathcal{E}_{u,v}$ holds, 0 otherwise
- $X := \sum_{\{u, v\} \in E} (X_{u,v} + X_{v,u})$
- **Claim:** $X \leq 2 \cdot$ **# deleted edges**
  - For each node v, at most one incident edges satisfies $\mathcal{E}_{u,v}$
  - so every edge incident to v is counted once, **for v**



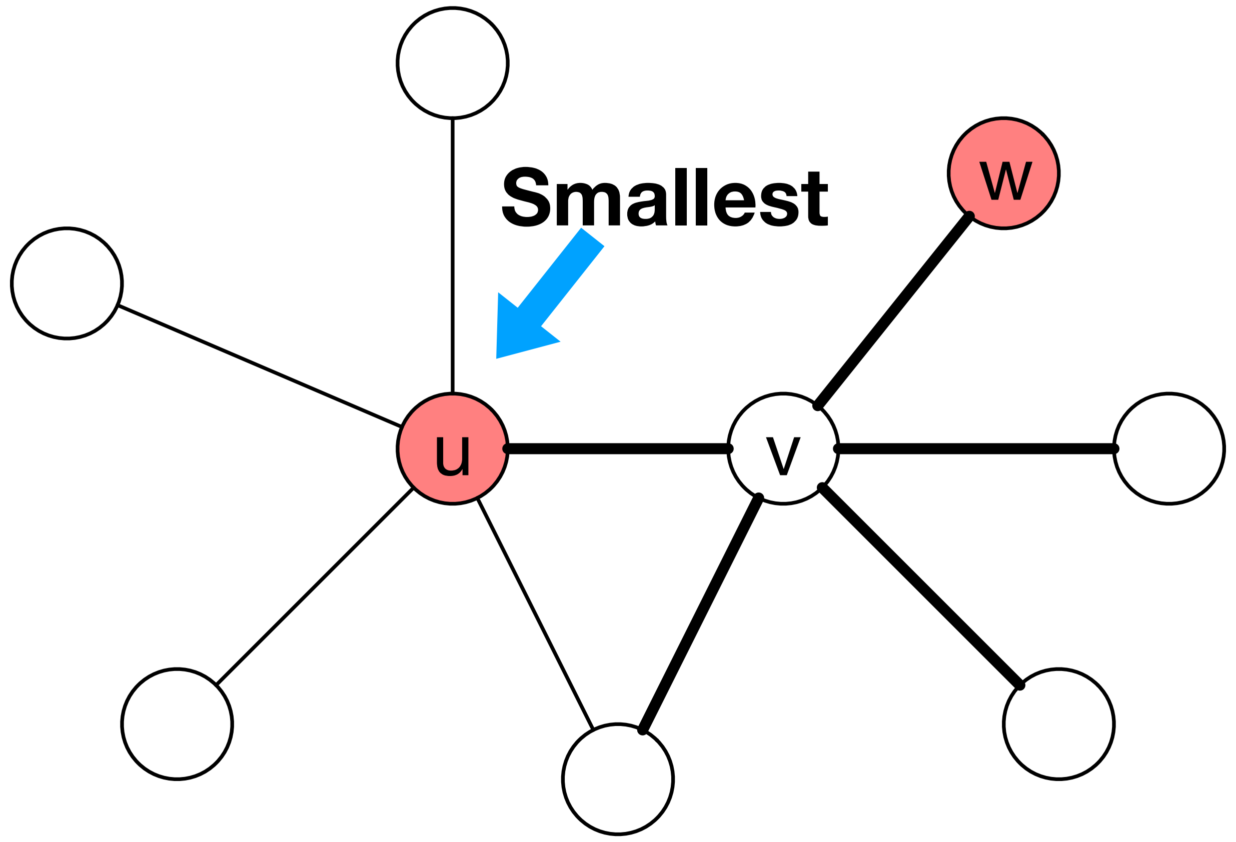
Randomized MIS: Luby's Algorithm

- $\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$
- $X_{u,v} := \deg(v)$ if $\mathcal{E}_{u,v}$ holds, 0 otherwise
- $X := \sum_{\{u,v\} \in E} (X_{u,v} + X_{v,u})$
- **Claim:** $X \leq 2 \cdot$ # deleted edges
  - For each node v, at most one incident edges satisfies $\mathcal{E}_{u,v}$
  - so every edge incident to v is counted once, **for v**
  - every edge can be **counted once**, **for each endpoint**



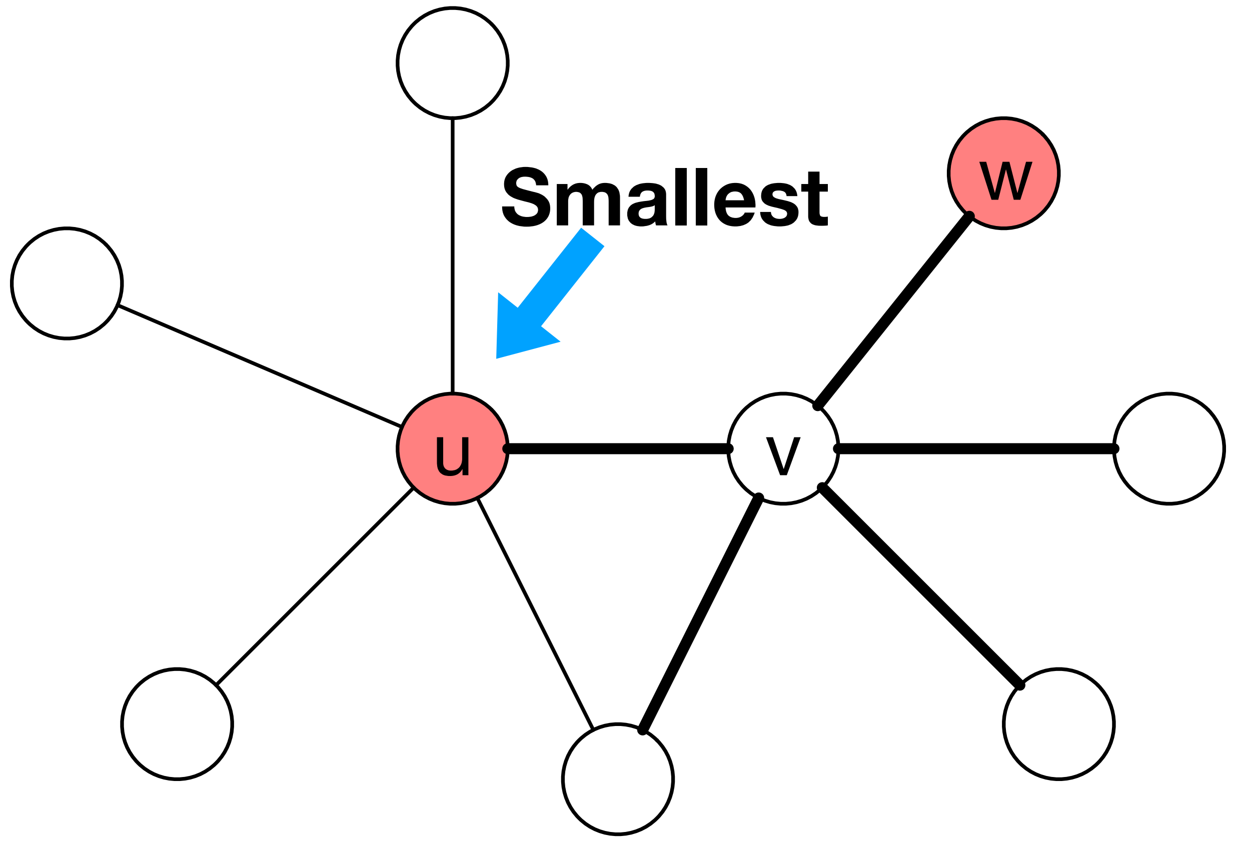
Randomized MIS: Luby's Algorithm

- $\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$
- $X_{u,v} :=$ deg(v) if $\mathcal{E}_{u,v}$ holds, 0 otherwise
- $X := \sum_{\{u, v\} \in E} (X_{u,v} + X_{v,u})$
- $X \leq 2 \cdot$ # deleted edges



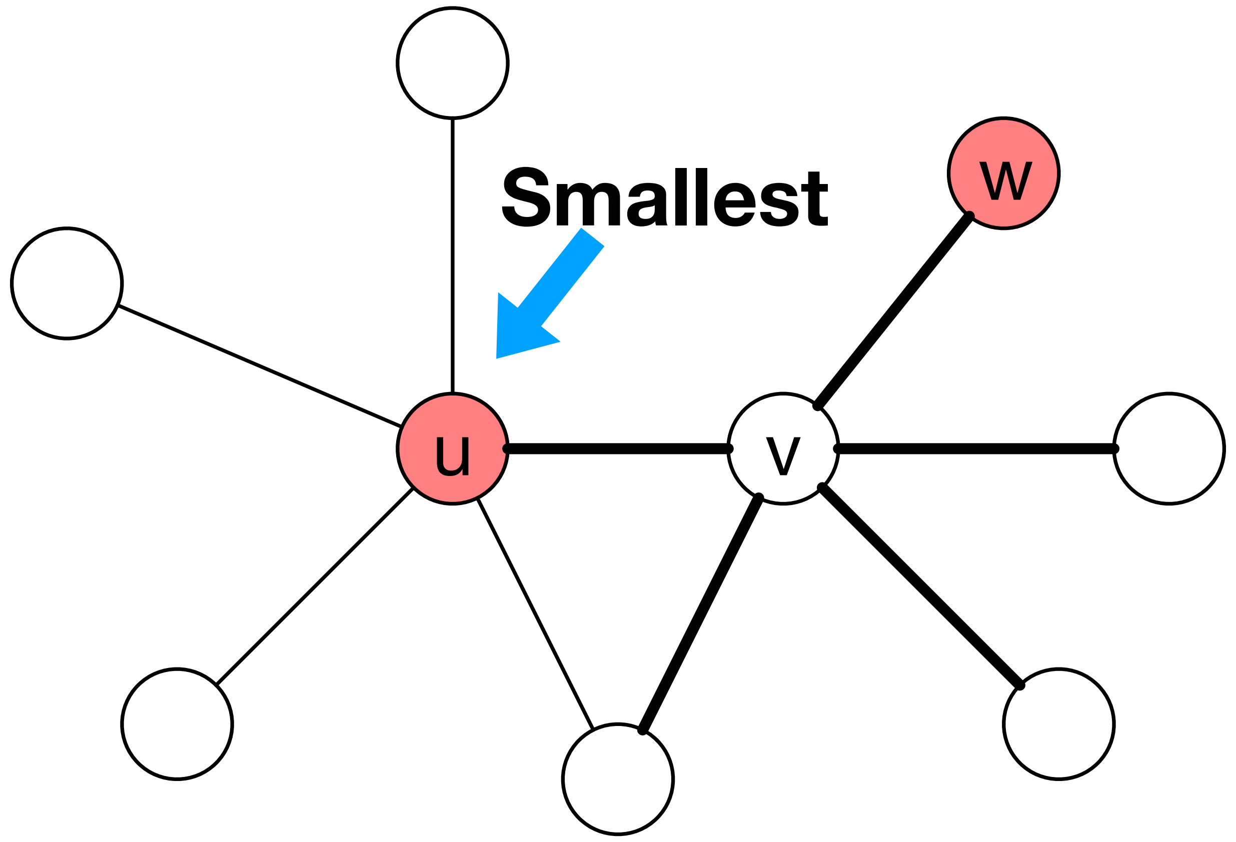
Randomized MIS: Luby's Algorithm

- $\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$
- $X_{u,v} := \deg(v)$ if $\mathcal{E}_{u,v}$ holds, 0 otherwise
- $X := \sum_{\{u,v\} \in E} (X_{u,v} + X_{v,u})$
- $X \leq 2 \cdot$ # deleted edges
- **Claim:** $\mathbb{E}[X] \geq |E|$



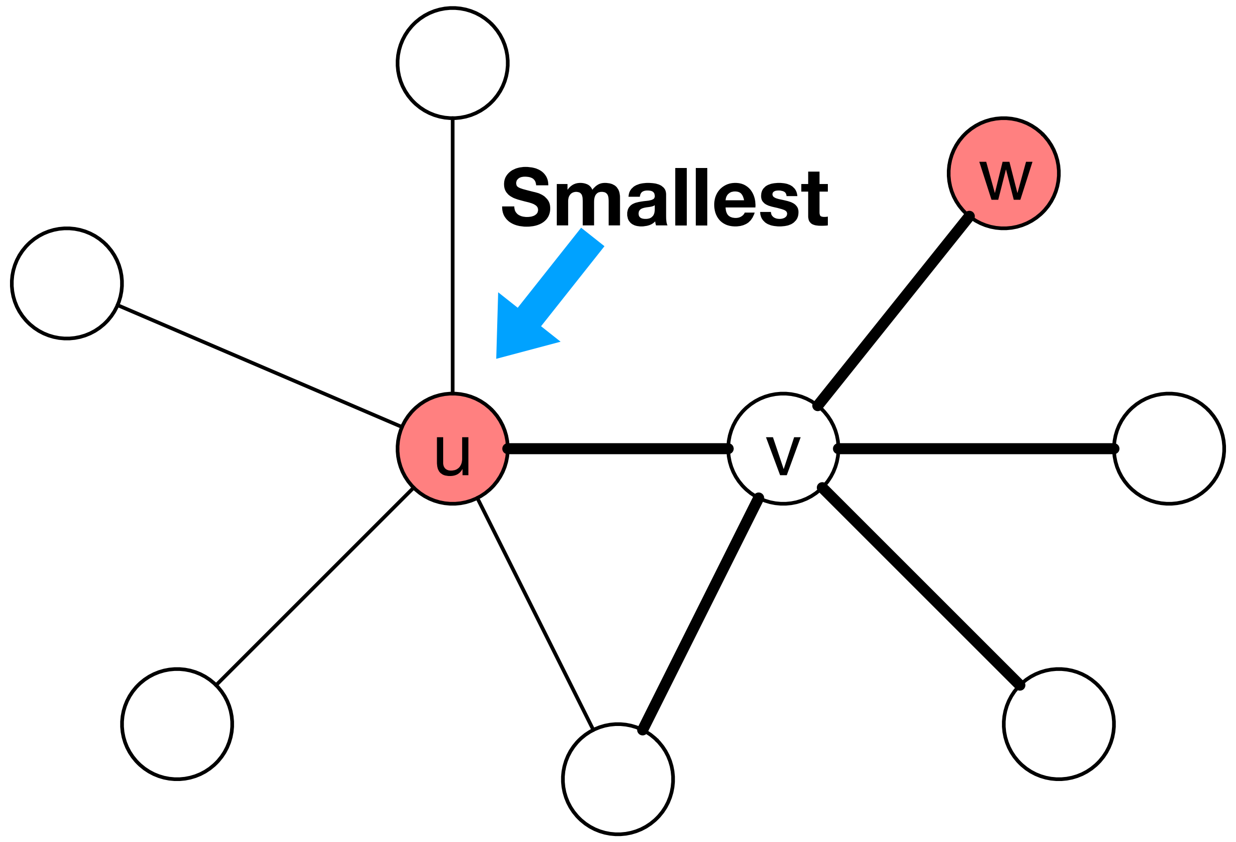
Randomized MIS: Luby's Algorithm

- $\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$
- $X_{u,v} := \deg(v)$ if $\mathcal{E}_{u,v}$ holds, 0 otherwise
- $X := \sum_{\{u, v\} \in E} (X_{u,v} + X_{v,u})$
- $X \leq 2 \cdot$ # deleted edges
- **Claim:** $\mathbb{E}[X] \geq |E|$
  - $\mathbb{E}[X_{u,v}] =$



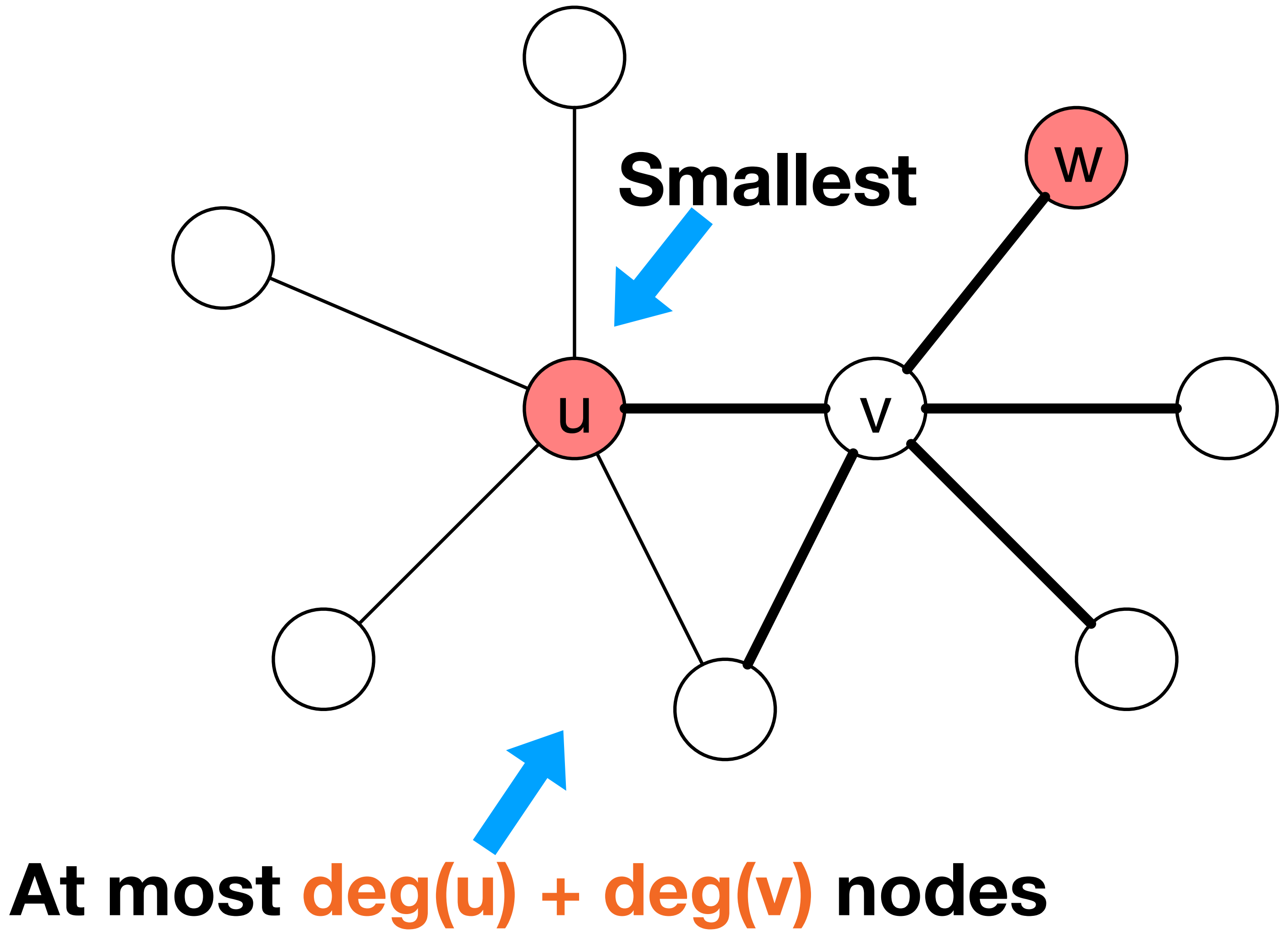
Randomized MIS: Luby's Algorithm

- $\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$
- $X_{u,v} :=$ deg(v) if $\mathcal{E}_{u,v}$ holds, 0 otherwise
- $X := \sum_{\{u,v\} \in E} (X_{u,v} + X_{v,u})$
- $X \leq 2 \cdot$ # deleted edges
- **Claim:** $\mathbb{E}[X] \geq |E|$
  - $\mathbb{E}[X_{u,v}] =$ deg(v) $\cdot$ $P(\mathcal{E}_{u,v})$



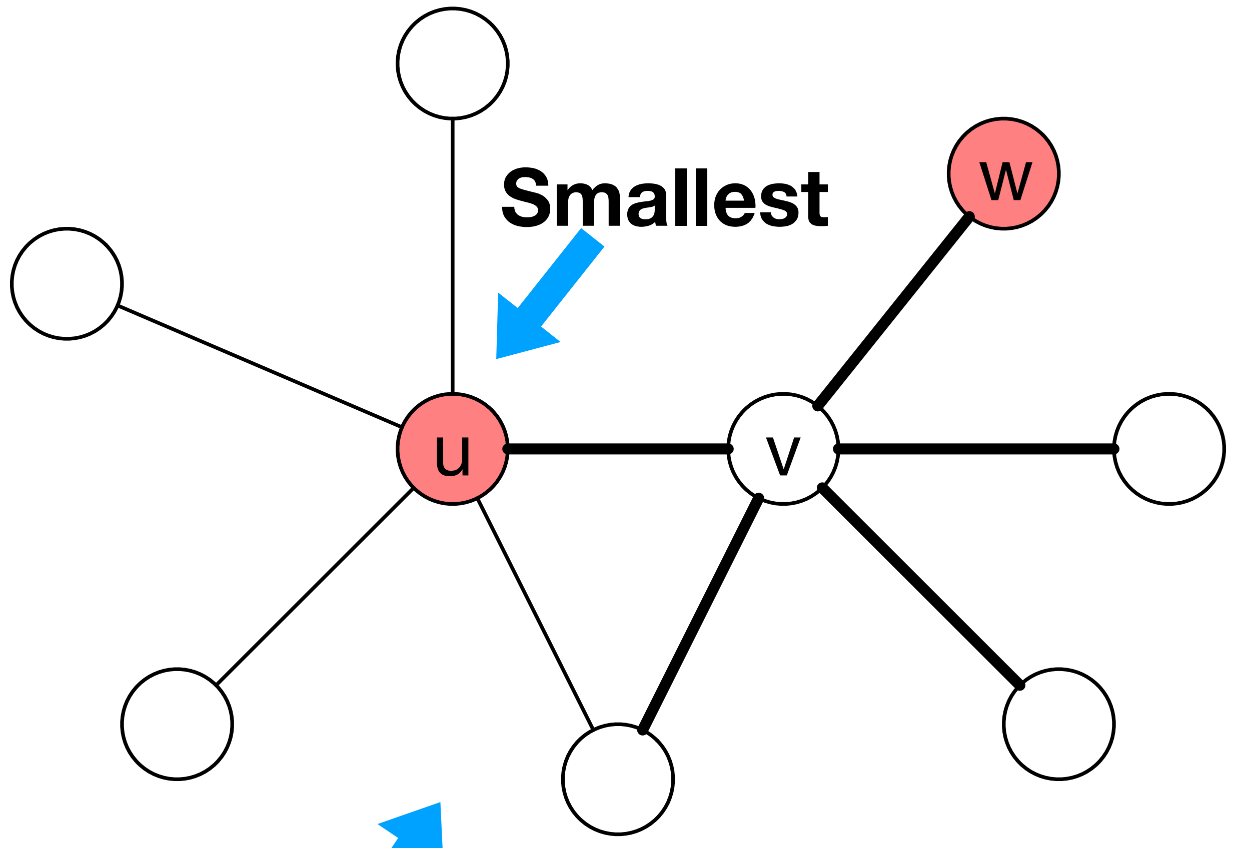
Randomized MIS: Luby's Algorithm

- $\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$
- $X_{u,v} := \deg(v)$ if $\mathcal{E}_{u,v}$ holds, 0 otherwise
- $X := \sum_{\{u,v\} \in E} (X_{u,v} + X_{v,u})$
- $X \leq 2 \cdot$ # deleted edges
- **Claim:** $\mathbb{E}[X] \geq |E|$
  - $\mathbb{E}[X_{u,v}] = \deg(v) \cdot P(\mathcal{E}_{u,v})$



Randomized MIS: Luby's Algorithm

- $\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$
- $X_{u,v} := \mathrm{deg}(v)$ if $\mathcal{E}_{u,v}$ holds, 0 otherwise
- $X := \sum_{\{u, v\} \in E} (X_{u,v} + X_{v,u})$
- $X \leq 2 \cdot$ # deleted edges
- **Claim:** $\mathbb{E}[X] \geq |E|$
  - $\mathbb{E}[X_{u,v}] = \mathrm{deg}(v) \cdot P(\mathcal{E}_{u,v}) \geq \mathrm{deg}(v) \cdot 1 / (\mathrm{deg}(u) + \mathrm{deg}(v))$

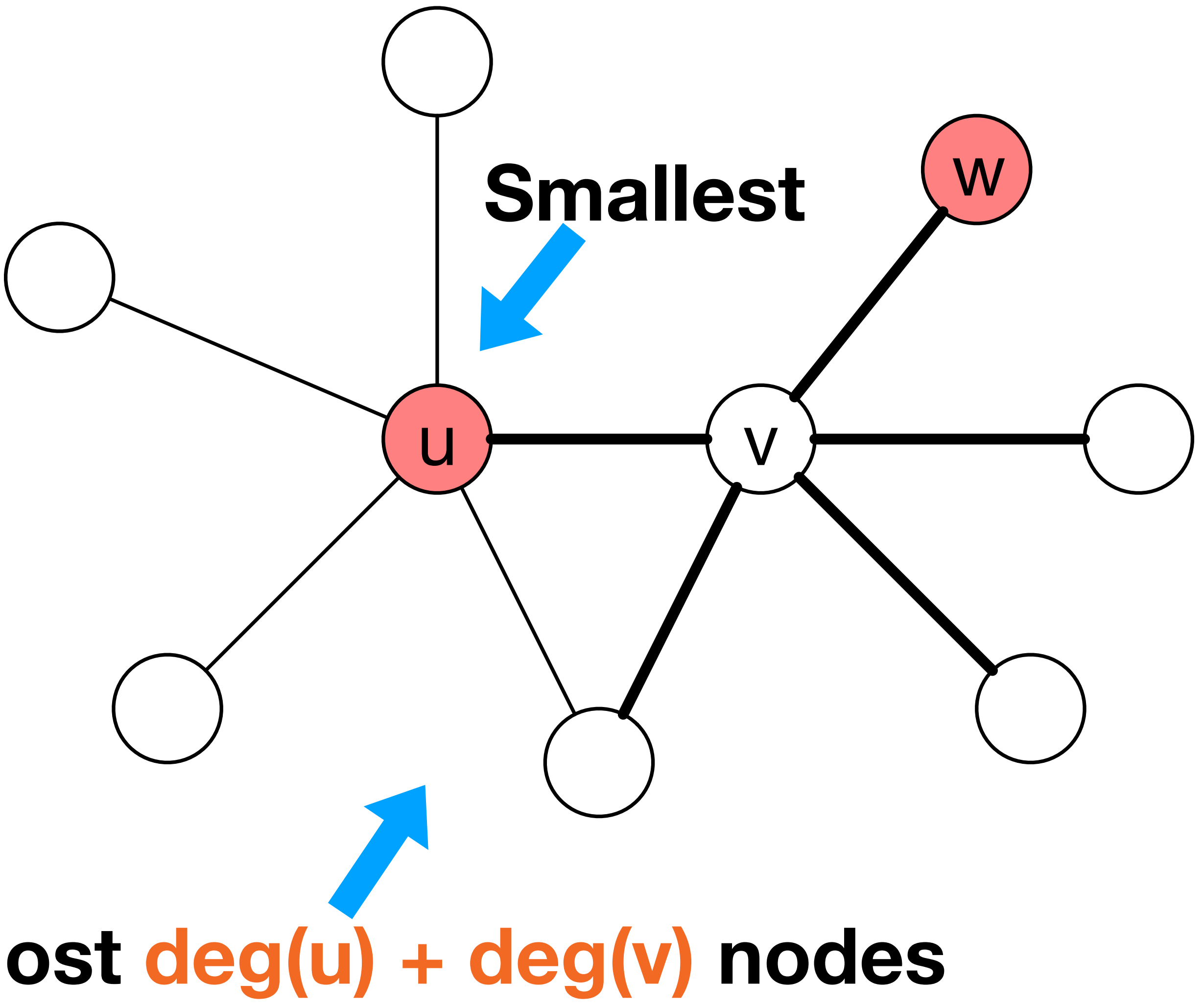


Smallest

At most deg(u) + deg(v) nodes

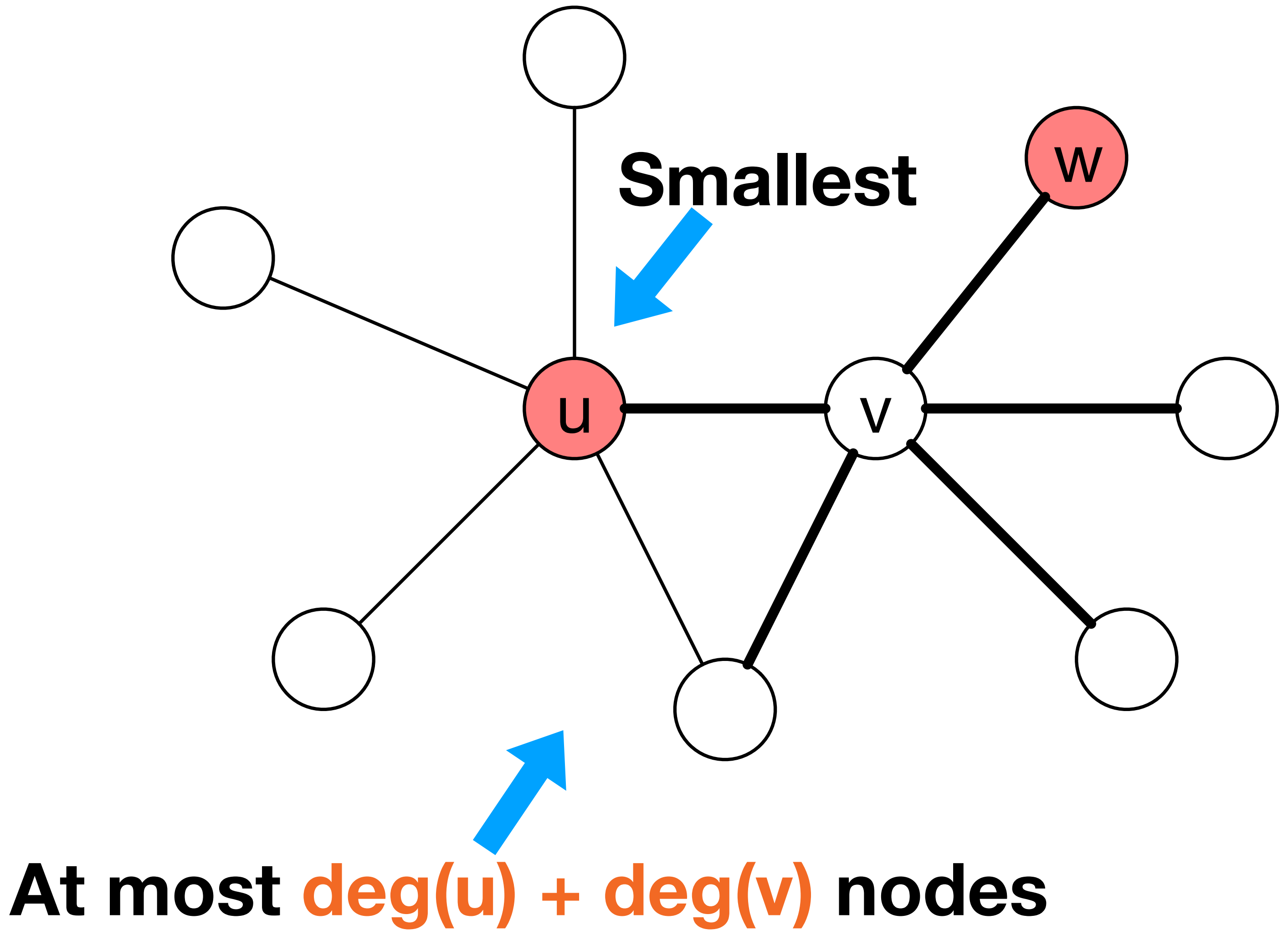
Randomized MIS: Luby's Algorithm

- $\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$
- $X_{u,v} := \deg(v)$ if $\mathcal{E}_{u,v}$ holds, 0 otherwise
- $X := \sum_{\{u,v\} \in E} (X_{u,v} + X_{v,u})$
- $X \leq 2 \cdot$ # deleted edges
- **Claim:** $\mathbb{E}[X] \geq |E|$
  - $\mathbb{E}[X_{u,v}] = \deg(v) \cdot P(\mathcal{E}_{u,v}) \geq \deg(v) \cdot 1 / (\deg(u) + \deg(v)) = \deg(v) / (\deg(u) + \deg(v))$



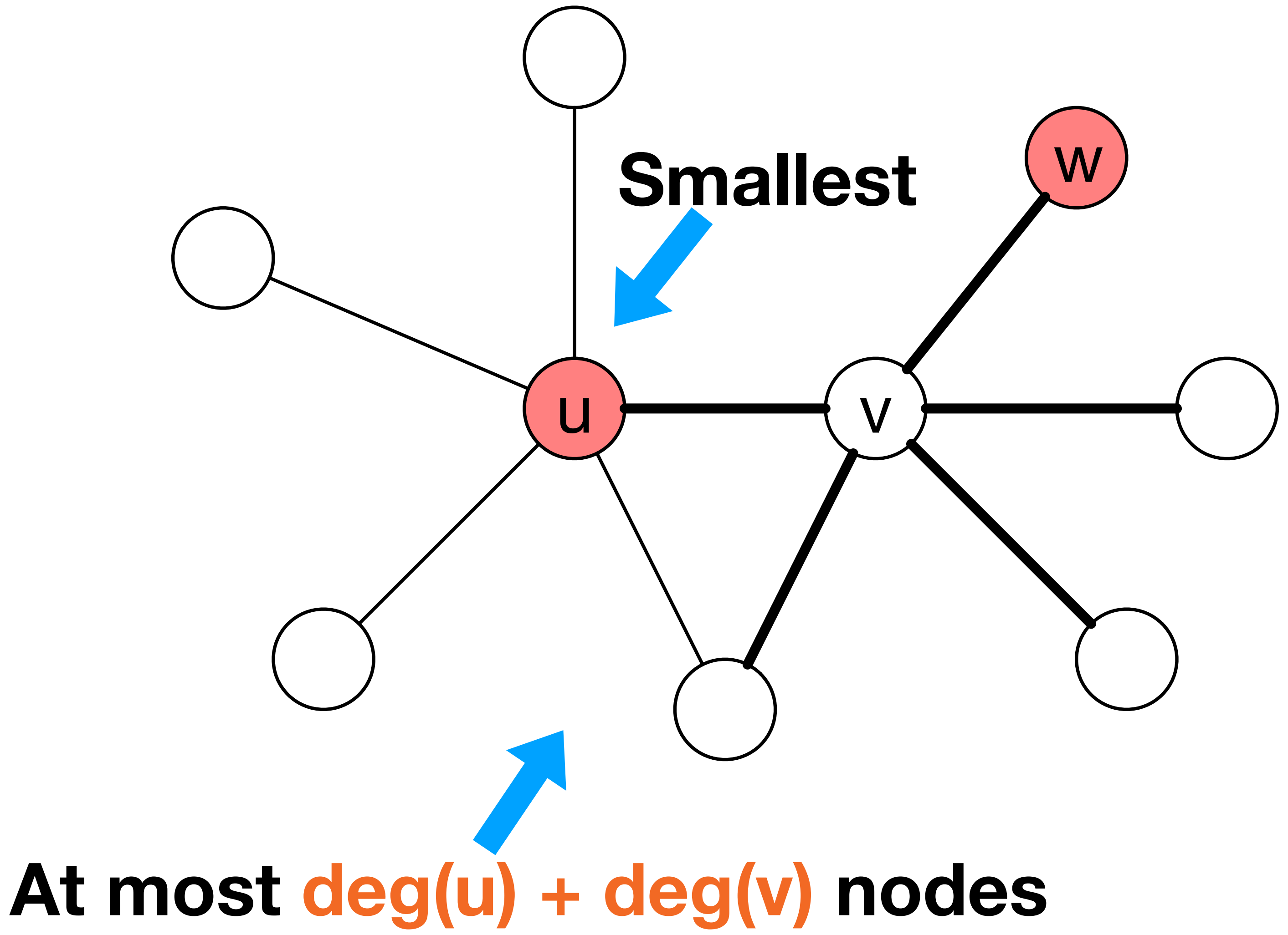
Randomized MIS: Luby's Algorithm

- $\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$
- $X_{u,v} := \deg(v)$ if $\mathcal{E}_{u,v}$ holds, 0 otherwise
- $X := \sum_{\{u, v\} \in E} (X_{u,v} + X_{v,u})$
- $X \leq 2 \cdot$ # deleted edges
- **Claim:** $\mathbb{E}[X] \geq |E|$
  - $\mathbb{E}[X_{u,v}] = \deg(v) \cdot P(\mathcal{E}_{u,v}) \geq \deg(v) \cdot 1 / (\deg(u) + \deg(v)) = \deg(v) / (\deg(u) + \deg(v))$
  - $\mathbb{E}[X] =$



Randomized MIS: Luby's Algorithm

- $\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$
- $X_{u,v} := \deg(v)$ if $\mathcal{E}_{u,v}$ holds, 0 otherwise
- $X := \sum_{\{u, v\} \in E} (X_{u,v} + X_{v,u})$
- $X \leq 2 \cdot$ # deleted edges

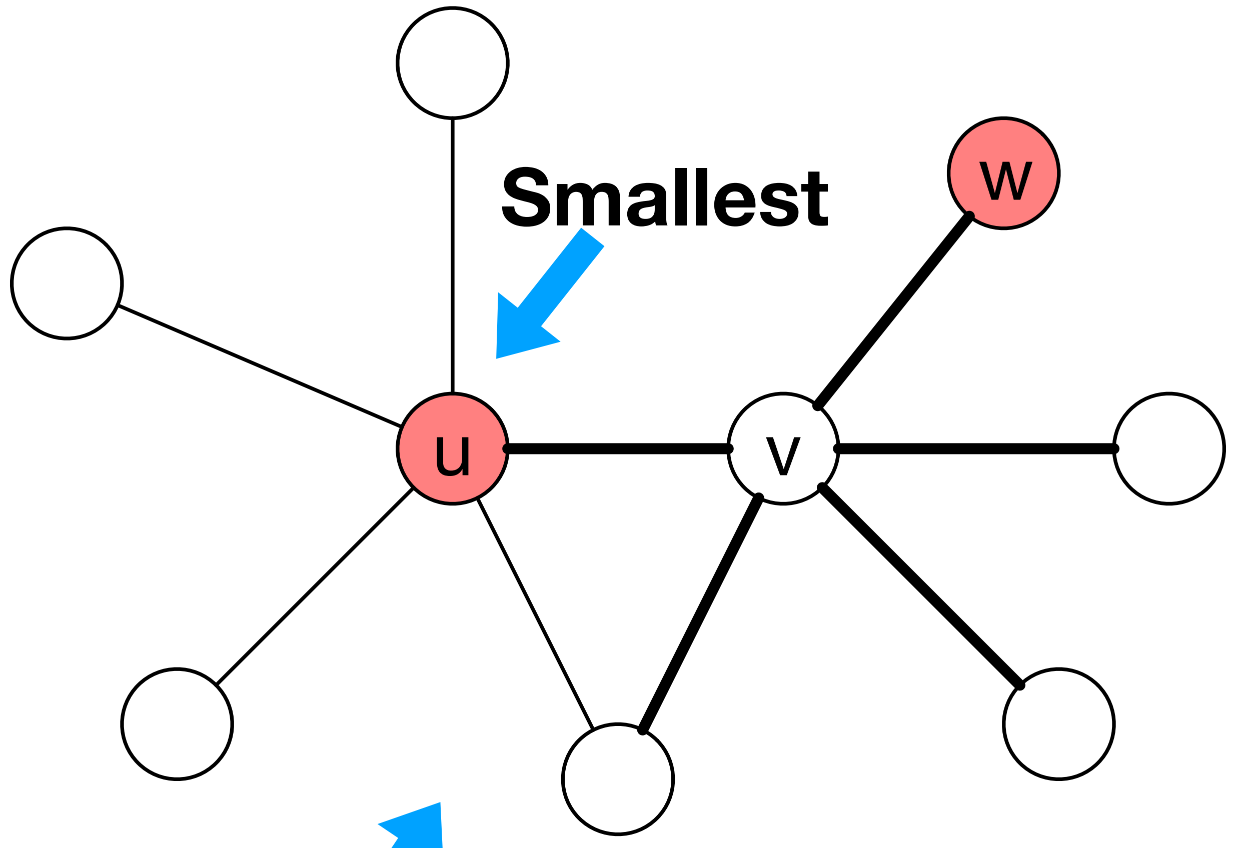


At most deg(u) + deg(v) nodes

- **Claim:** $\mathbb{E}[X] \geq |E|$
  - $\mathbb{E}[X_{u,v}] = \deg(v) \cdot P(\mathcal{E}_{u,v}) \geq \deg(v) \cdot 1 / (\deg(u) + \deg(v)) = \deg(v) / (\deg(u) + \deg(v))$
  - $\mathbb{E}[X] = \mathbb{E}\Big[\sum_{\{u,v\} \in E} (X_{u,v} + X_{v,u})\Big]$

Randomized MIS: Luby's Algorithm

- $\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$
- $X_{u,v} :=$ deg(v) if $\mathcal{E}_{u,v}$ holds, 0 otherwise
- $X := \sum_{\{u,v\} \in E} (X_{u,v} + X_{v,u})$
- $X \leq 2 \cdot$ # deleted edges



Smallest

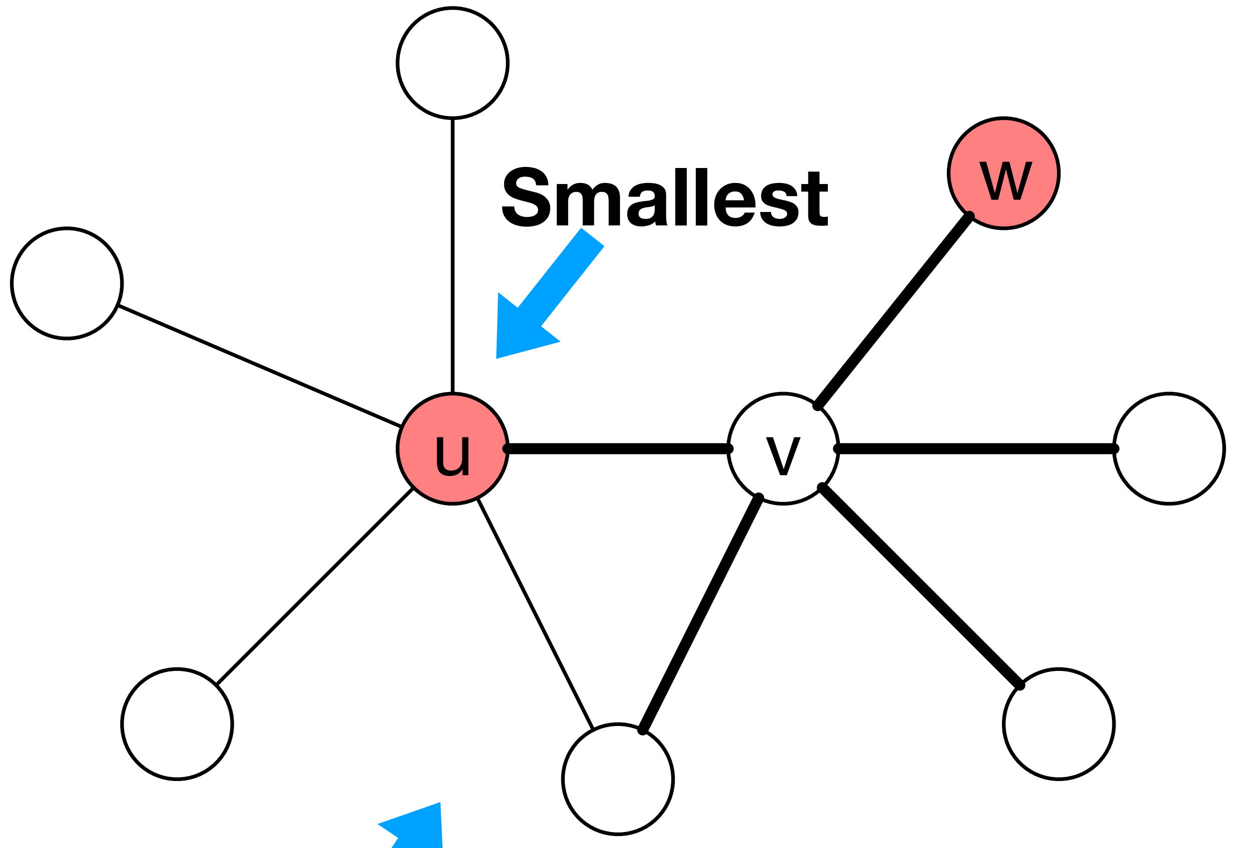
At most deg(u) + deg(v) nodes

- **Claim:** $\mathbb{E}[X] \geq |E|$
  - $\mathbb{E}[X_{u,v}] = \text{deg}(v) \cdot P(\mathcal{E}_{u,v}) \geq \text{deg}(v) \cdot 1 / (\text{deg}(u) + \text{deg}(v)) = \text{deg}(v) / (\text{deg}(u) + \text{deg}(v))$
  - $\mathbb{E}[X] = \mathbb{E}\left[\sum_{\{u,v\} \in E} (X_{u,v} + X_{v,u}) \right]$

Linearity of expectation: $\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y]$

Randomized MIS: Luby's Algorithm

- $\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$
- $X_{u,v} := deg(v)$ if $\mathcal{E}_{u,v}$ holds, 0 otherwise
- $X := \sum_{\{u, v\} \in E} (X_{u,v} + X_{v,u})$
- $X \leq 2 \cdot$ # deleted edges



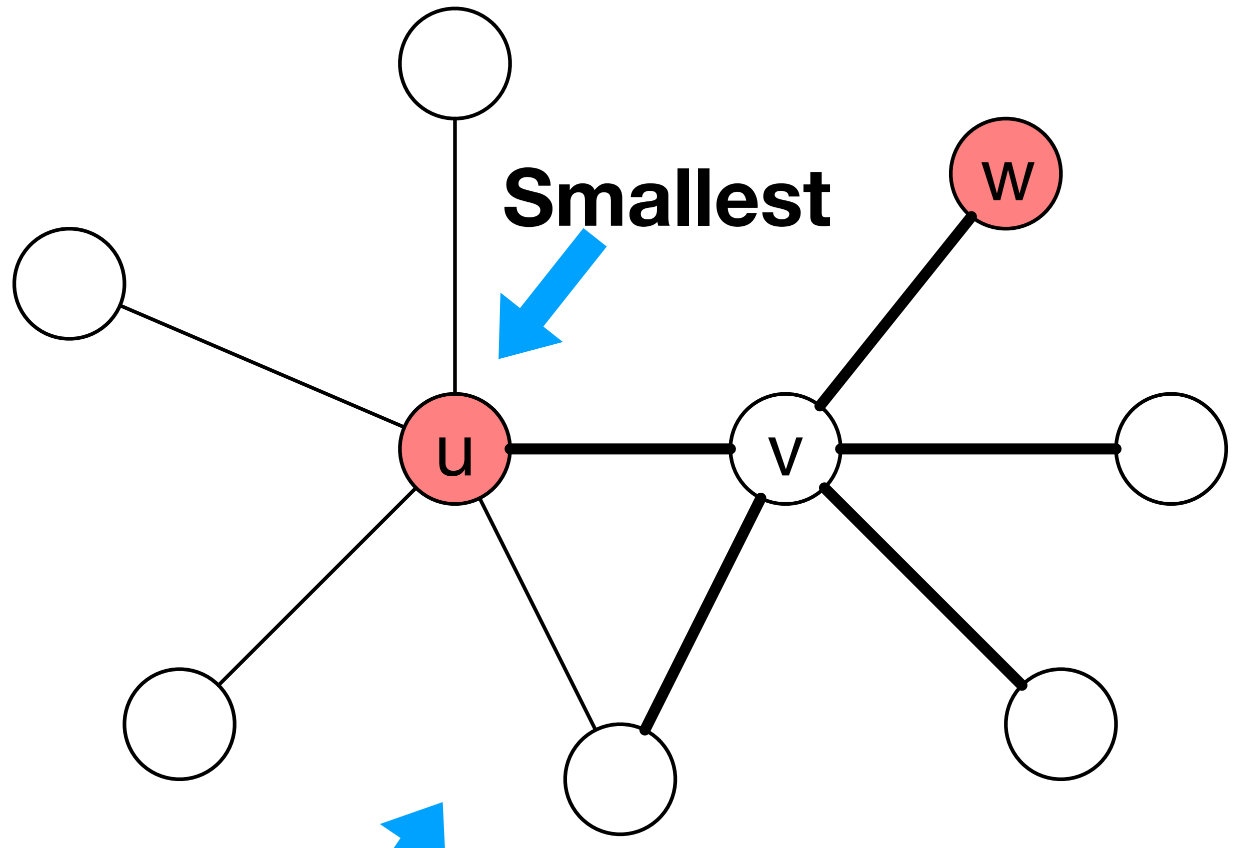
At most deg(u) + deg(v) nodes

- Claim: $\mathbb{E}[X] \geq |E|$
  - $\mathbb{E}[X_{u,v}] = deg(v) \cdot P(\mathcal{E}_{u,v}) \geq deg(v) \cdot 1 / (deg(u) + deg(v)) = deg(v) / (deg(u) + deg(v))$
  - $\mathbb{E}[X] = \mathbb{E}\left[\sum_{\{u,v\} \in E} (X_{u,v} + X_{v,u}) \right] = \sum_{\{u,v\} \in E} \left(\mathbb{E}[X_{u,v}] + \mathbb{E}[X_{v,u}] \right)$

Linearity of expectation: $\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y]$

Randomized MIS: Luby's Algorithm

- $\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$
- $X_{u,v} := \deg(v)$ if $\mathcal{E}_{u,v}$ holds, 0 otherwise
- $X := \sum_{\{u,v\} \in E} (X_{u,v} + X_{v,u})$
- $X \leq 2 \cdot$ # deleted edges
- **Claim:** $\mathbb{E}[X] \geq |E|$
  - $\mathbb{E}[X_{u,v}] = \deg(v) \cdot P(\mathcal{E}_{u,v}) \geq \deg(v) \cdot 1 / (\deg(u) + \deg(v)) = \deg(v) / (\deg(u) + \deg(v))$
  - $\mathbb{E}[X] = \mathbb{E}\Big[\sum_{\{u,v\} \in E} (X_{u,v} + X_{v,u}) \Big] = \sum_{\{u,v\} \in E} \Big(\mathbb{E}[X_{u,v}] + \mathbb{E}[X_{v,u}] \Big) \geq \sum_{\{u,v\} \in E} 1$

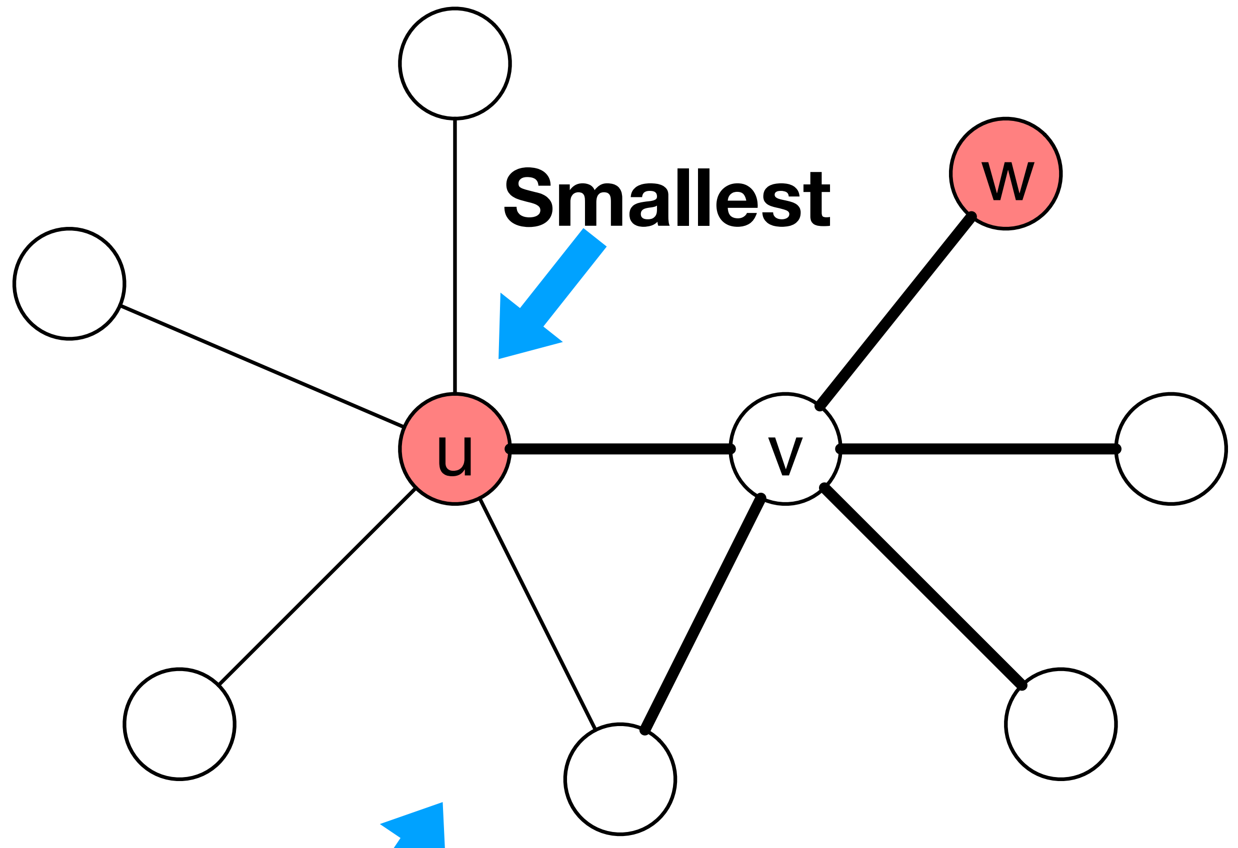


At most deg(u) + deg(v) nodes

Linearity of expectation: $\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y]$

Randomized MIS: Luby's Algorithm

- $\mathcal{E}_{u,v} \Leftrightarrow \forall w \in N(u) \cup N(v) \setminus \{u\} : X_u < X_w$
- $X_{u,v} := \deg(v)$ if $\mathcal{E}_{u,v}$ holds, 0 otherwise
- $X := \sum_{\{u,v\} \in E} (X_{u,v} + X_{v,u})$
- $X \leq 2 \cdot$ # deleted edges
- **Claim:** $\mathbb{E}[X] \geq |E|$
  - $\mathbb{E}[X_{u,v}] = \deg(v) \cdot P(\mathcal{E}_{u,v}) \geq \deg(v) \cdot 1 / (\deg(u) + \deg(v)) = \deg(v) / (\deg(u) + \deg(v))$
  - $\mathbb{E}[X] = \mathbb{E}\left[\sum_{\{u,v\}\in E}(X_{u,v} + X_{v,u})\right] = \sum_{\{u,v\}\in E}\left(\mathbb{E}[X_{u,v}] + \mathbb{E}[X_{v,u}]\right) \geq \sum_{\{u,v\}\in E} 1 = |E|$



At most $\deg(u) + \deg(v)$ **nodes**

Linearity of expectation: $\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y]$

Randomized MIS: Luby's Algorithm

- **Lemma:** in **expectation**, at least **half** of the remaining **edges** are **removed**.

Randomized MIS: Luby's Algorithm

- **Lemma:** in **expectation**, at least **half** of the remaining **edges** are **removed**.
- **X** ≤ **2** · **# deleted edges**

Randomized MIS: Luby's Algorithm

- **Lemma:** in **expectation**, at least **half** of the remaining **edges** are **removed**.
- $X \leq 2 \cdot$ **# deleted edges**
- $\mathbb{E}[X] \geq |E|$

Randomized MIS: Luby's Algorithm

- **Lemma:** in **expectation**, at least **half** of the remaining **edges** are **removed**.
- **$X \le 2 \cdot$ # deleted edges**
- **$\mathbb{E}[X] \ge |E|$**
- **# deleted edges $\ge$ $X / 2$**

Randomized MIS: Luby's Algorithm

- **Lemma:** in **expectation**, at least **half** of the remaining **edges** are **removed**.
- $X \leq 2 \cdot \#$ deleted edges
- $\mathbb{E}[X] \geq |E|$
- $\#$ deleted edges $\geq X / 2$
- $\mathbb{E}[\, \#$ deleted edges $] \geq \mathbb{E}[\, X / 2 \,] \geq |E| / 2$

Randomized MIS: Luby's Algorithm

- **Lemma:** in **expectation**, at least **half** of the remaining **edges** are **removed**.

Randomized MIS: Luby's Algorithm

- **Lemma:** in **expectation**, at least **half** of the remaining **edges** are **removed**.
- **Theorem:** Luby's randomized MIS algorithm computes an MIS in time **O(log n)** in expectation and with high probability.

Randomized MIS: Luby's Algorithm

- **Lemma:** in **expectation**, at least **half** of the remaining **edges** are **removed**.
- **Theorem:** Luby's randomized MIS algorithm computes an MIS in time **O(log n)** in expectation and with high probability.

Randomized MIS: Luby's Algorithm

- **Lemma:** in **expectation**, at least **half** of the remaining **edges** are **removed**.

- **Theorem:** Luby's randomized MIS algorithm computes an MIS in time **$O(\log n)$** in expectation and with high probability.

  - **Markov's inequality**: **$P(X \geq a) \leq \mathbb{E}[X] / a$**

Randomized MIS: Luby's Algorithm

- **Lemma:** in **expectation**, at least **half** of the remaining **edges** are **removed**.
- **Theorem:** Luby's randomized MIS algorithm computes an MIS in time **O(log n)** in expectation and with high probability.

- **Markov's inequality**: $P(X \geq a) \leq \mathbb{E}[X] / a$
- What is the **probability** that the number of **remaining edges** is at least a fraction **3/4**?

Randomized MIS: Luby's Algorithm

- **Lemma:** in **expectation**, at least **half** of the remaining **edges** are **removed**.
- **Theorem:** Luby's randomized MIS algorithm computes an MIS in time **$O(\log n)$** in expectation and with high probability.

- **Markov's inequality**: **$P(X \geq a) \leq \mathbb{E}[X] / a$**
- What is the **probability** that the number of **remaining edges** is at least a fraction **3/4**?
- By Markov: at most **2/3**

Randomized MIS: Luby's Algorithm

- **Lemma:** in **expectation**, at least **half** of the remaining **edges** are **removed**.
- **Theorem:** Luby's randomized MIS algorithm computes an MIS in time **O(log n)** in expectation and with high probability.

  - **Markov's inequality**: **$P(X \geq a) \leq \mathbb{E}[X] / a$**
  - What is the **probability** that the number of **remaining edges** is at least a fraction **3/4**?
  - By Markov: at most **2/3**
  - Hence, with **probability** at least **1/3** we **remove** of at least a fraction **1/4** of edges.

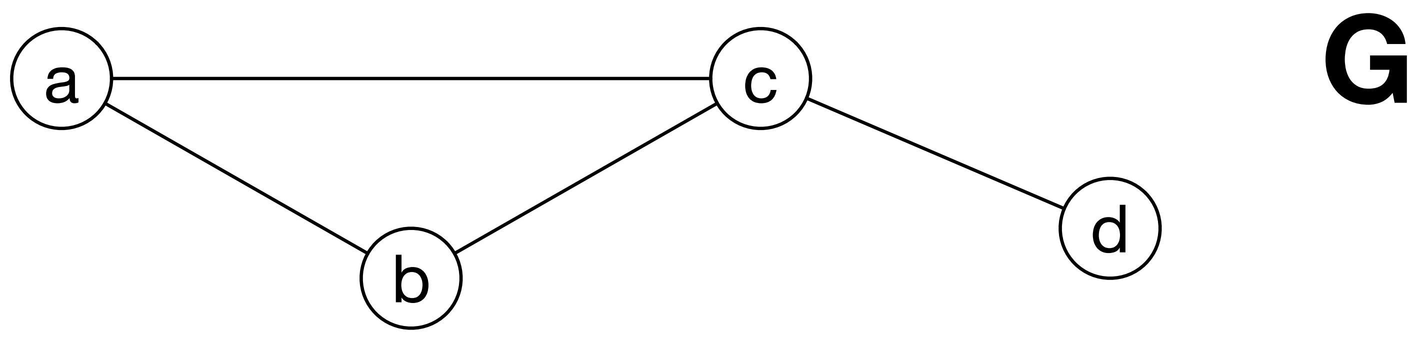
Randomized MIS: Luby's Algorithm

- **Lemma:** in **expectation**, at least **half** of the remaining **edges** are **removed**.
- **Theorem:** Luby's randomized MIS algorithm computes an MIS in time **O(log n)** in expectation and with high probability.

- **Markov's inequality**: **$P(X \geq a) \leq \mathbb{E}[X] / a$**
- What is the **probability** that the number of **remaining edges** is at least a fraction **3/4**?
- By Markov: at most **2/3**
- Hence, with **probability** at least **1/3** we **remove** of at least a fraction **1/4** of edges.

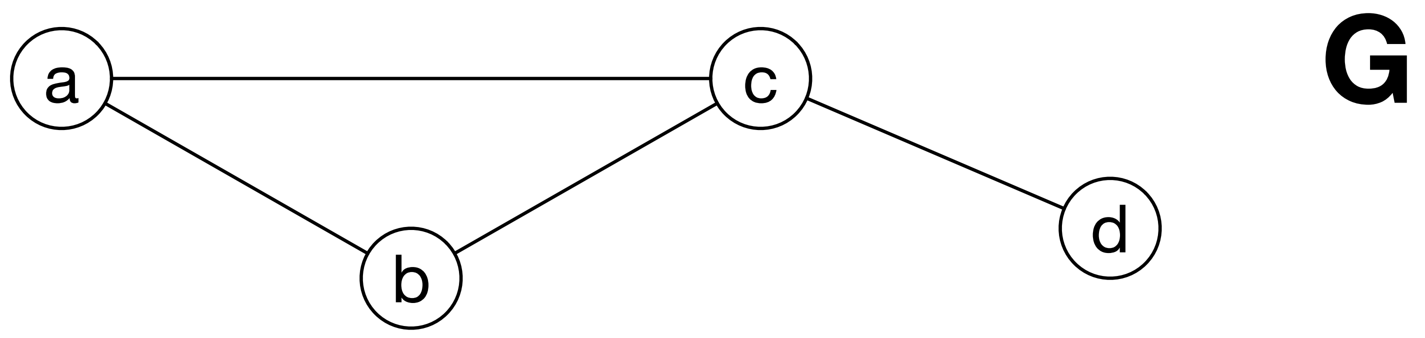
From MIS to (Δ+1)-coloring

From MIS to (Δ+1)-coloring



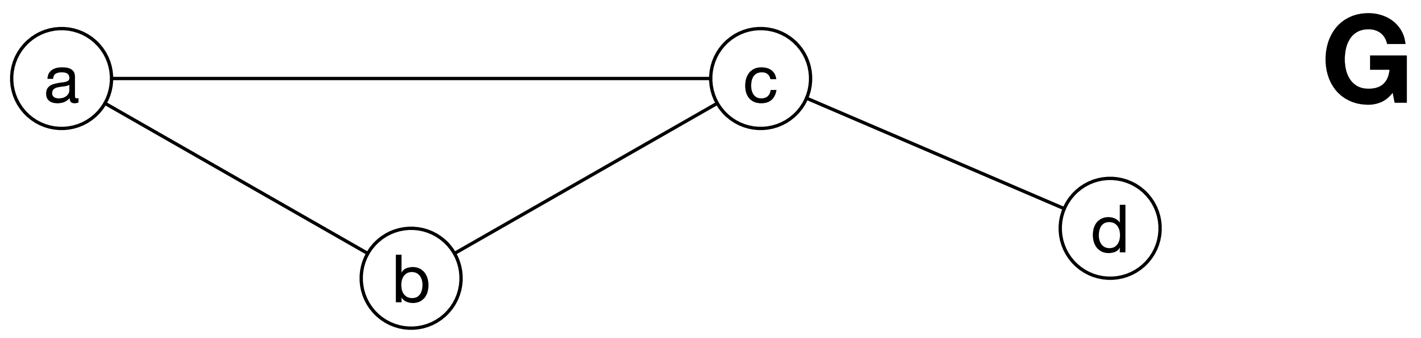
- Assume we want to compute a **coloring** on graph **G**

From MIS to (Δ+1)-coloring



- Assume we want to compute a coloring on graph G
- We transform G into a new virtual graph H

From MIS to (Δ+1)-coloring

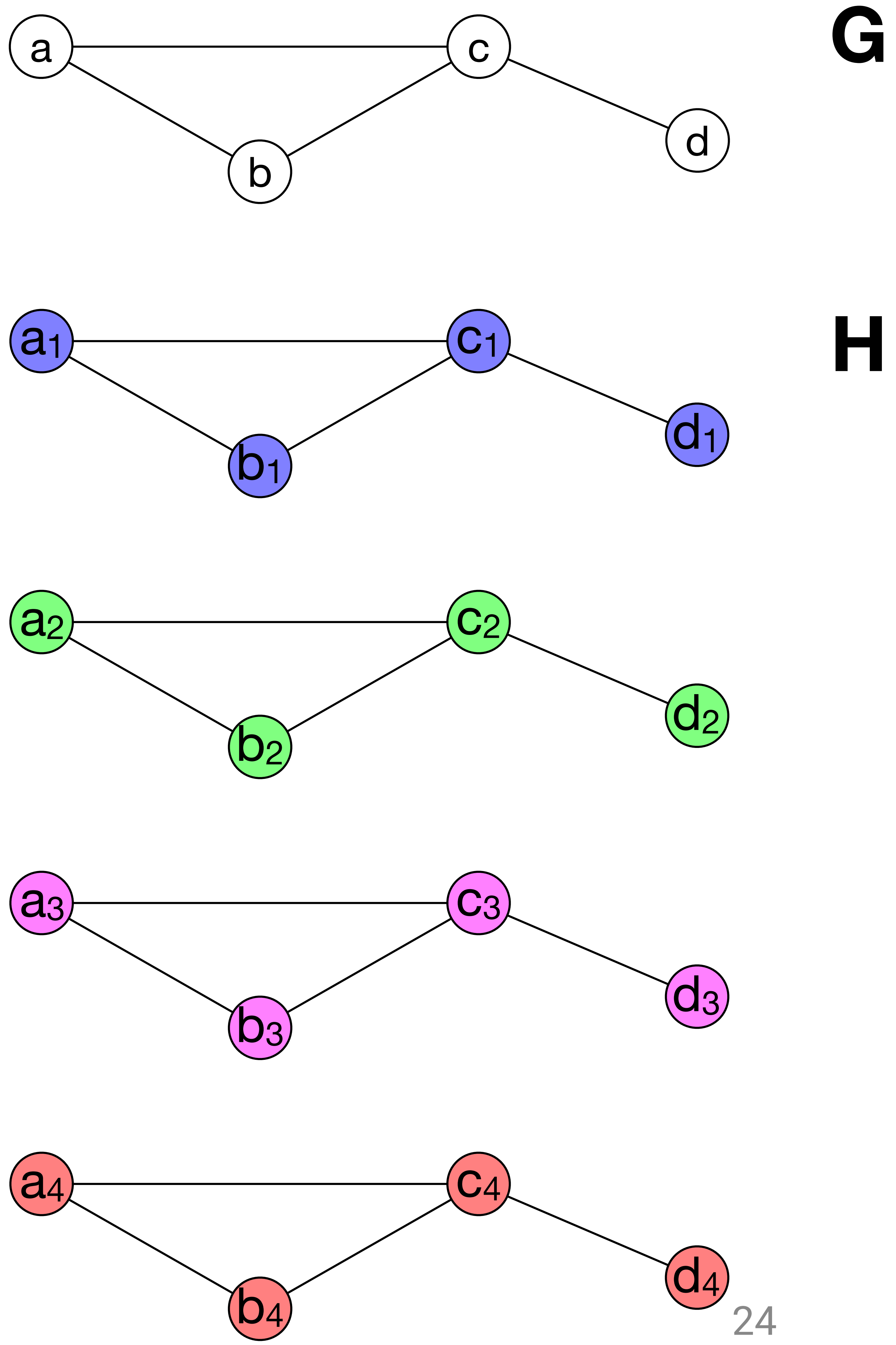


- Assume we want to compute a **coloring** on graph **G**
- We **transform** **G** into a new **virtual** graph **H**
  - that can be **simulated** on **G**

From MIS to (Δ+1)-coloring

- Assume we want to compute a **coloring** on graph **G**
- We **transform** **G** into a new **virtual** graph **H**
  - that can be **simulated** on **G**

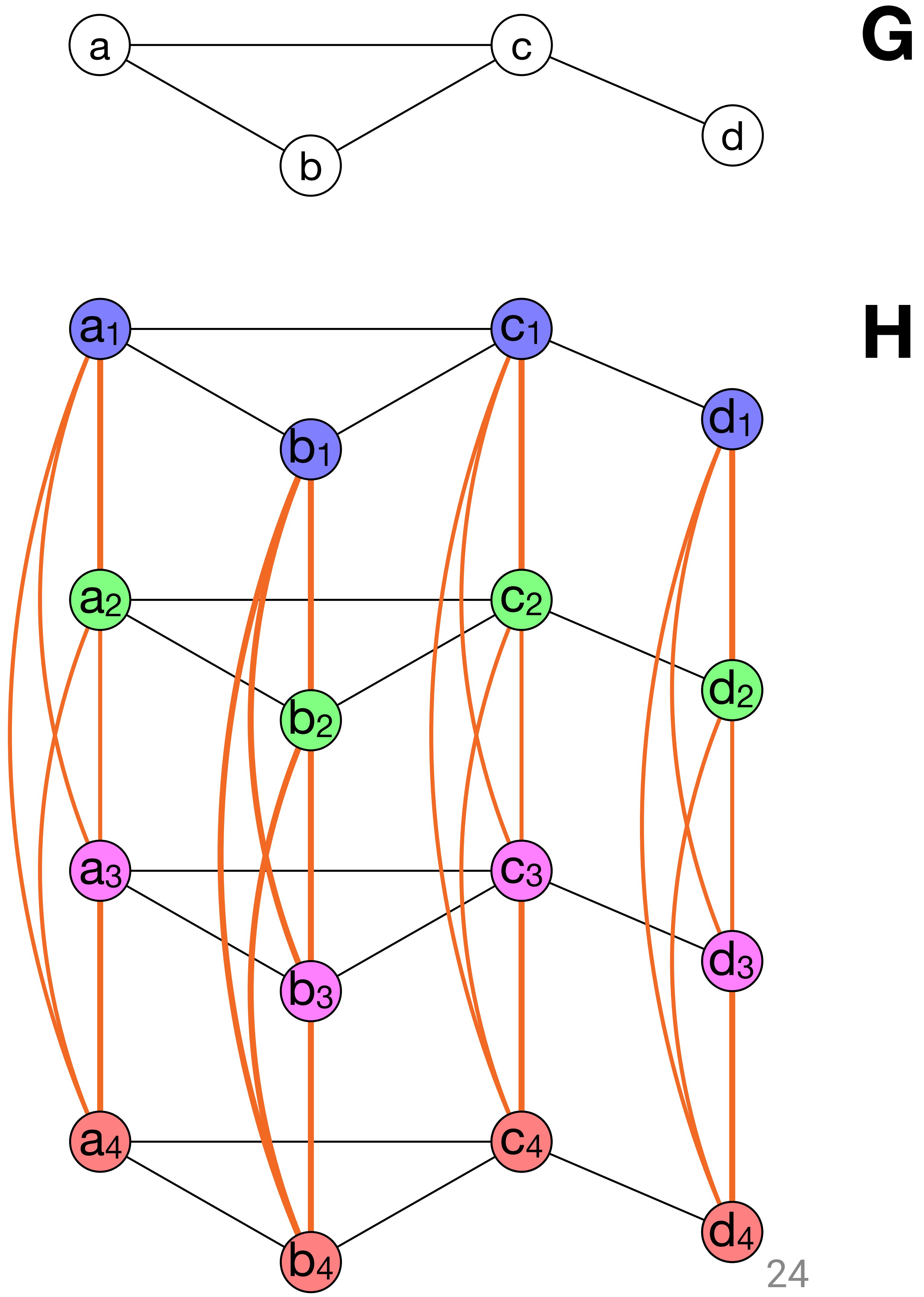
1. Create **Δ+1** copies of **G**



From MIS to (Δ+1)-coloring

- Assume we want to compute a **coloring** on graph **G**
- We **transform** **G** into a new **virtual** graph **H**
  - that can be **simulated** on **G**

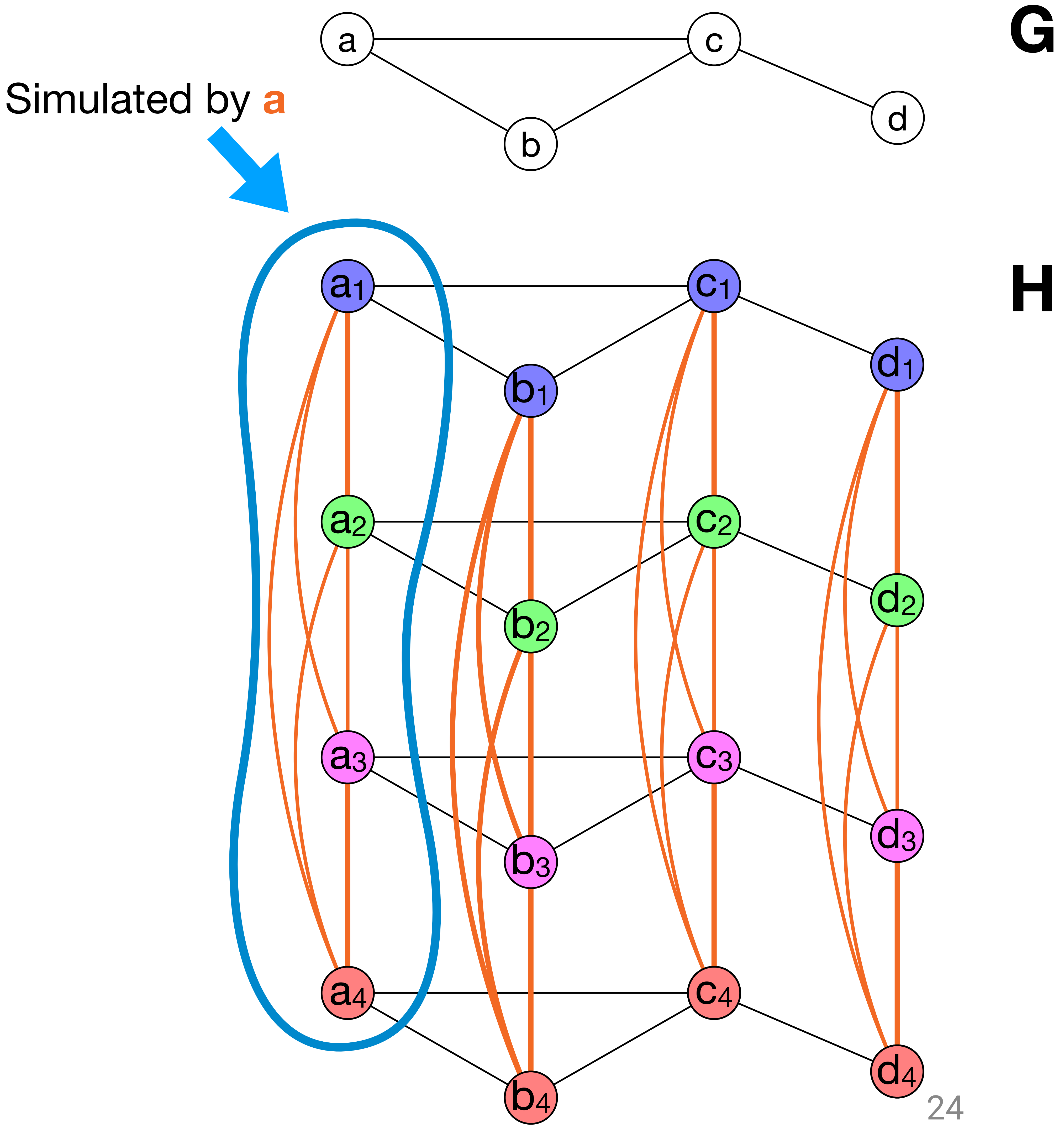
1. Create **Δ+1** copies of **G**
2. **Connect** corresponding nodes in the copies to a **clique**



From MIS to (Δ+1)-coloring

- Assume we want to compute a **coloring** on graph **G**
- We **transform** **G** into a new **virtual** graph **H**
    - that can be **simulated** on **G**

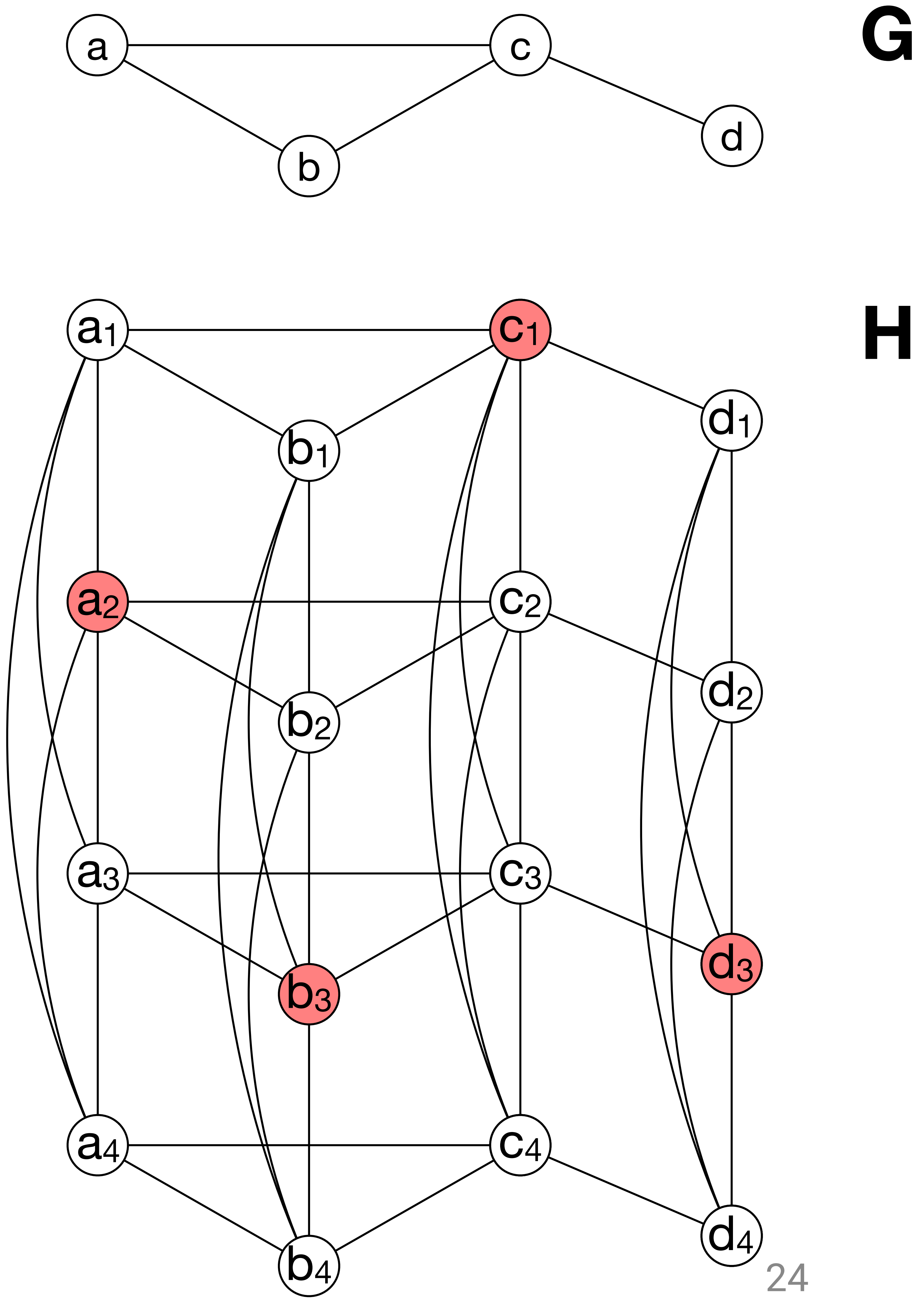
1. Create **Δ+1** copies of **G**
2. **Connect** corresponding nodes in the copies to a **clique**



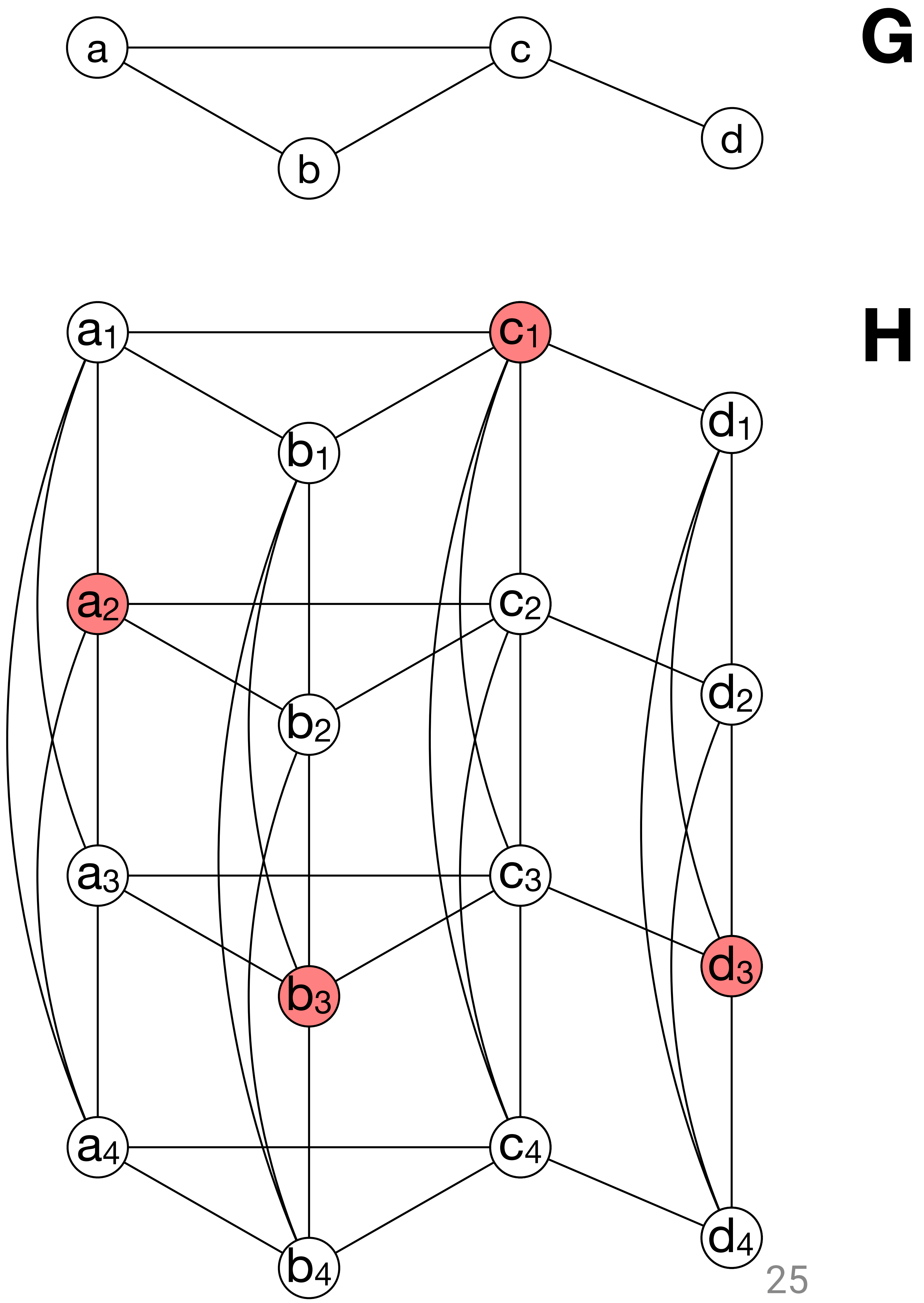
From MIS to (Δ+1)-coloring

- Assume we want to compute a **coloring** on graph **G**
- We **transform** **G** into a new **virtual** graph **H**
  - that can be **simulated** on **G**

1. Create **Δ+1** copies of **G**
2. **Connect** corresponding nodes in the copies to a **clique**
3. Compute **MIS** on **H**

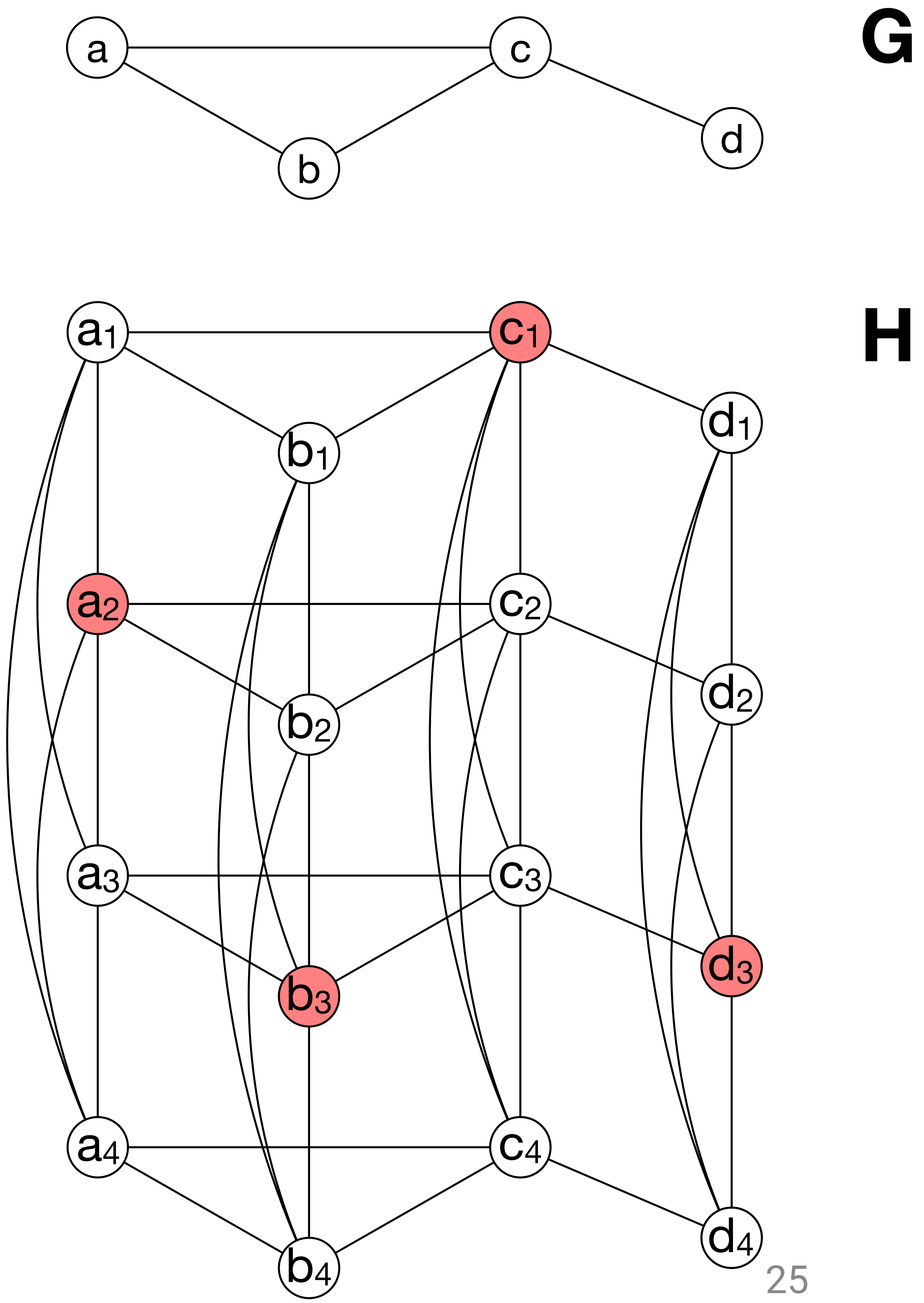


From MIS to (Δ+1)-coloring



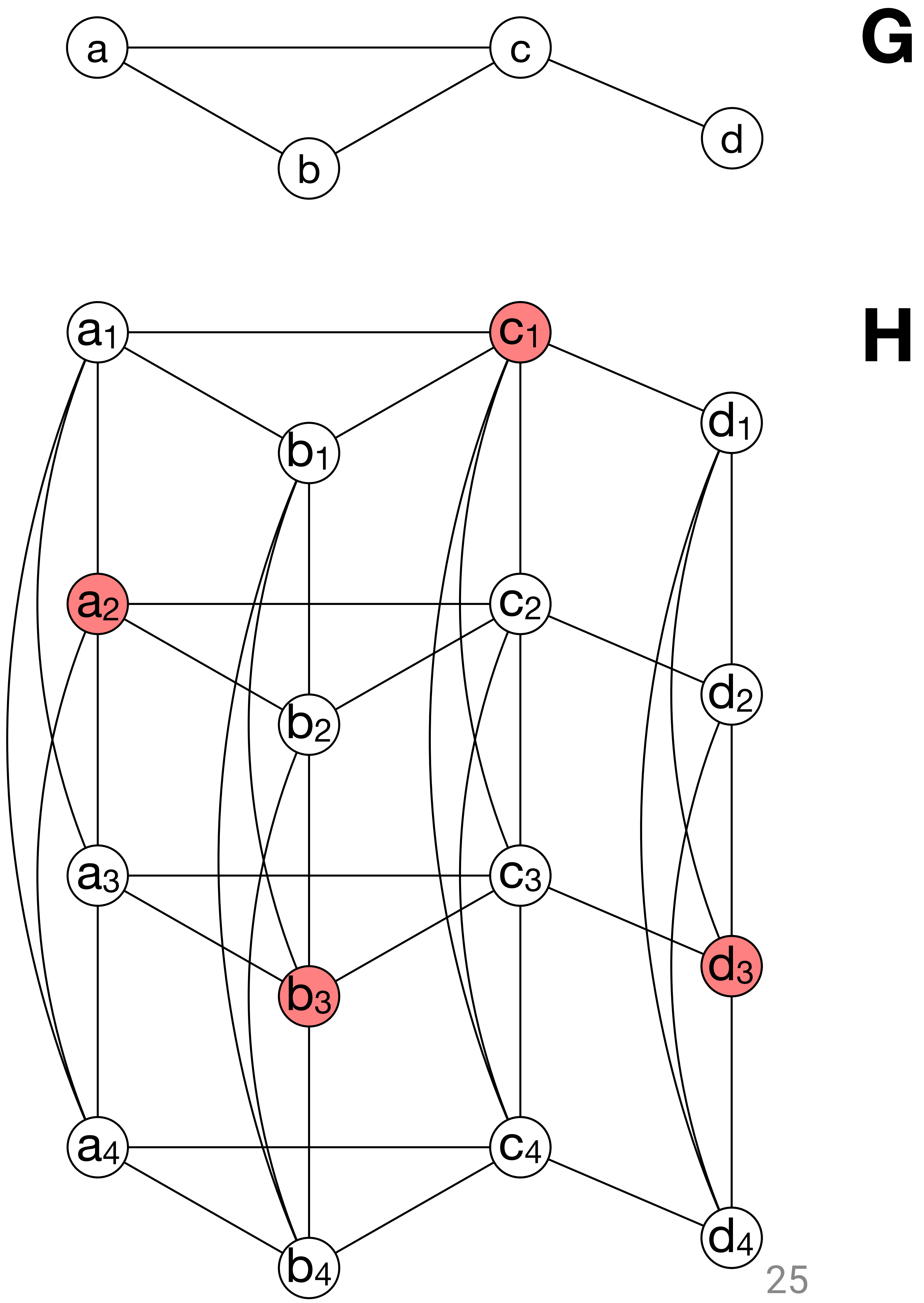
From MIS to (Δ+1)-coloring

- **Claim**: the MIS contains exactly one node from each column



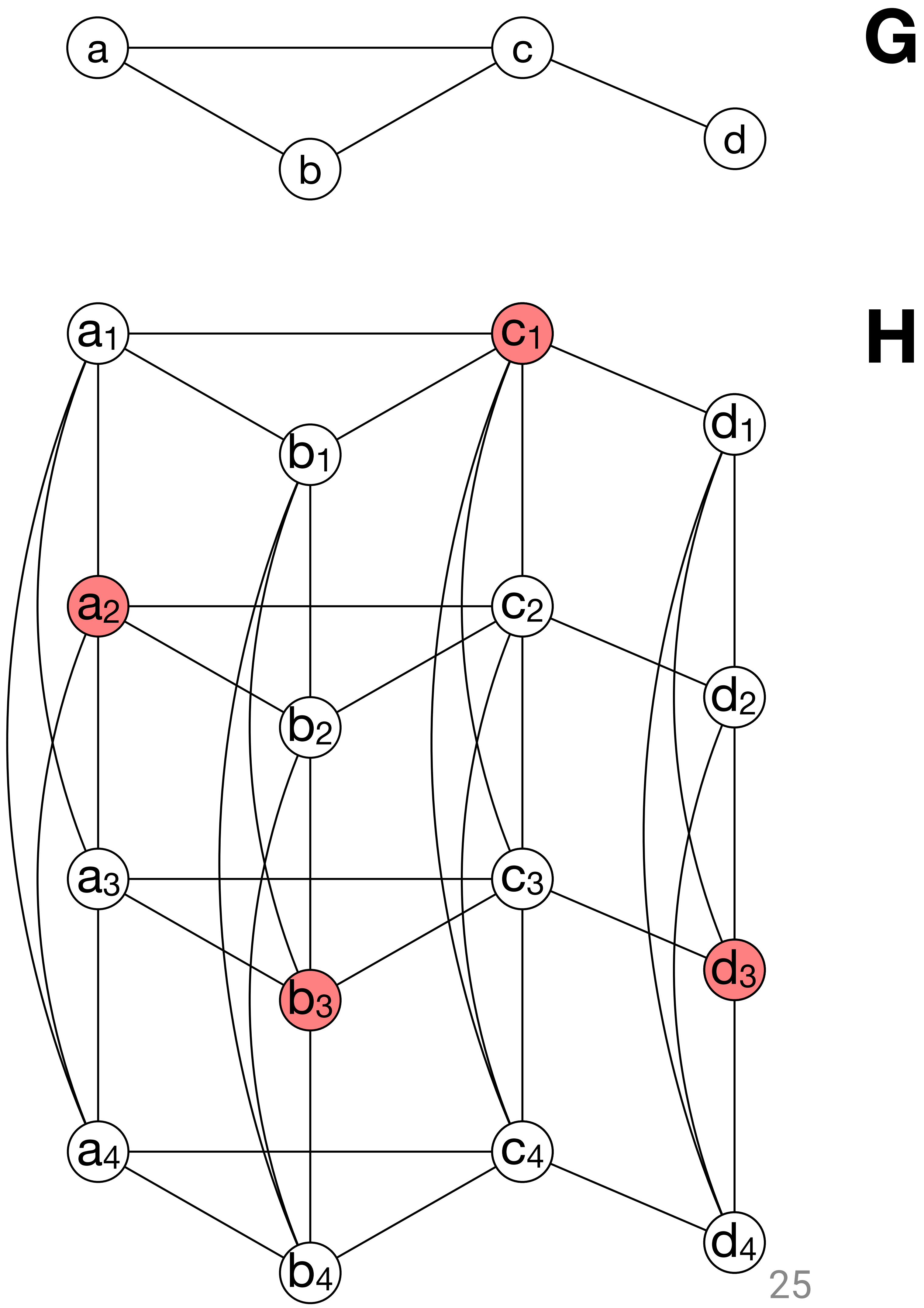
From MIS to (Δ+1)-coloring

- **Claim**: the MIS contains exactly one node from each column
  - **At most 1**: each column forms a clique



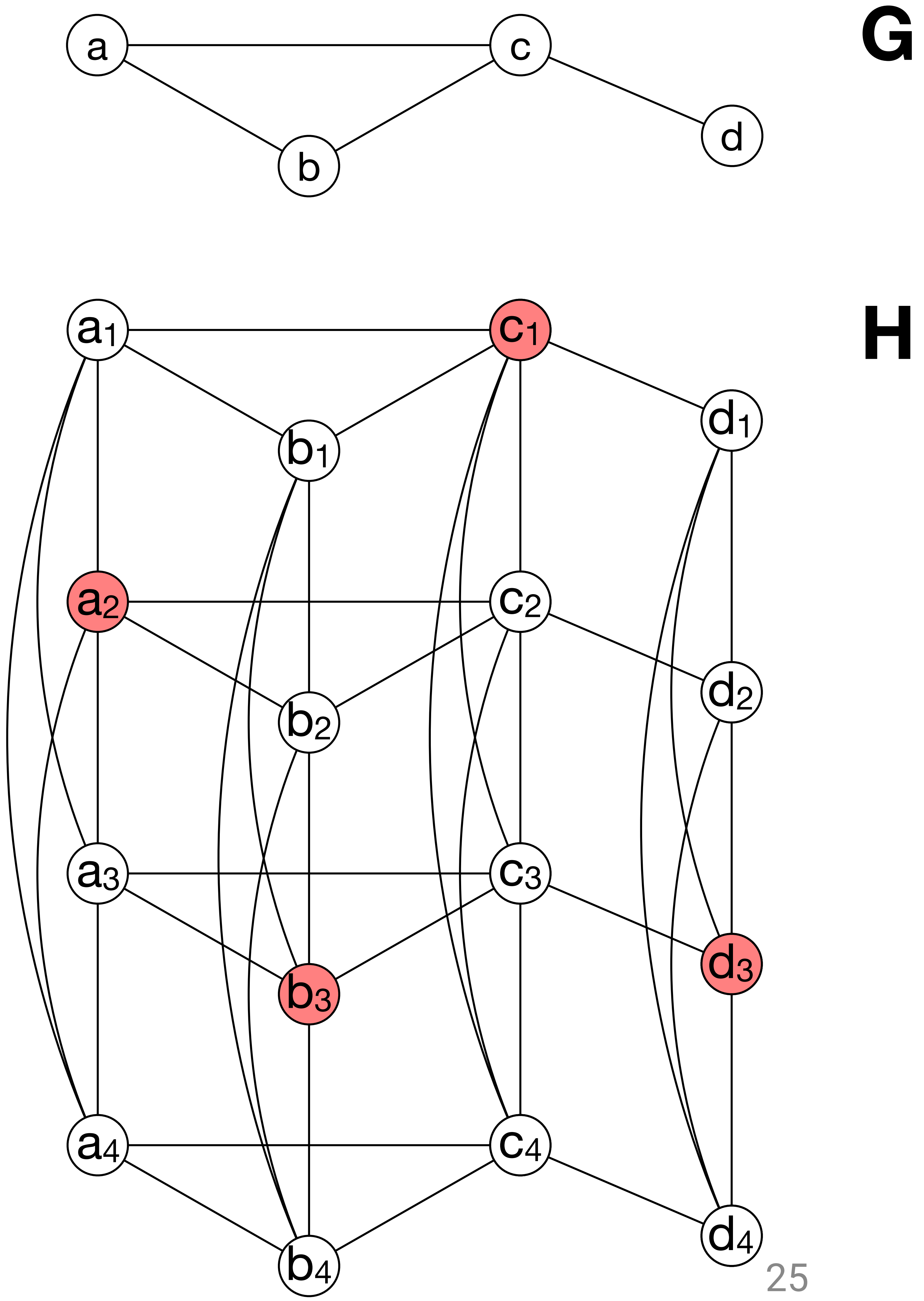
From MIS to (Δ+1)-coloring

- **Claim**: the MIS contains exactly one node from each column
  - **At most 1**: each column forms a clique
  - **At least 1**: each neighbor in G can cover at most 1 copy. But there are at most Δ neighbors in G, and Δ+1 copies. So at least one node for each column cannot be a neighbor of MIS nodes of other columns.



From MIS to (Δ+1)-coloring

- **Claim**: the MIS contains exactly one node from each column
  - **At most 1**: each column forms a clique
  - **At least 1**: each neighbor in **G** can **cover at most 1 copy**. But there are at most **Δ** neighbors in G, and **Δ+1** copies. So at least one node for each column cannot be a neighbor of MIS nodes of other columns.
- **Algorithm**: if a **column node v_i** is in the **MIS**, node v picks **color i**



From MIS to (Δ+1)-coloring

From MIS to (Δ+1)-coloring

- **Theorem**: together with the O(log n) randomized MIS algorithm, this reduction gives an alternative way to compute a Δ+1 coloring in O(log n) rounds.

From MIS to (Δ+1)-coloring

- **Theorem**: together with the O(log n) randomized MIS algorithm, this reduction gives an alternative way to compute a Δ+1 coloring in O(log n) rounds.

- **Remark**: this construction can be modified to assign colors in {1, ..., deg(v)+1}:

From MIS to (Δ+1)-coloring

- **Theorem**: together with the **O(log n)** randomized **MIS** algorithm, this reduction gives an alternative way to compute a **Δ+1** coloring in **O(log n)** rounds.

- **Remark**: this construction can be modified to assign colors in **{1, ..., deg(v)+1}**:
  - Put **deg(v)+1** copies of **v** instead of **Δ+1**

Summary

Summary

- **MIS and Δ+1 coloring (randomized):** **O(log n)** [Luby '86]

Summary

- **MIS and Δ+1 coloring (randomized):** O(log n) [Luby '86]
- **Best deterministic algorithms:**

Summary

- **MIS and Δ+1 coloring (randomized):** **$O(\log n)$** [Luby '86]
- **Best deterministic algorithms:**
  - **MIS:** **$O(\log^c n)$ rounds** [Rozhoň, Ghaffari, 2019]

Summary

- **MIS and Δ+1 coloring (randomized):** **$O(\log n)$** [Luby '86]
- **Best deterministic algorithms:**
  - **MIS:** **$O(\log^c n)$ rounds** [Rozhoň, Ghaffari, 2019]
  - **Δ+1 coloring:** **$O(\log^2 n \log \Delta)$ rounds** [Kuhn, Ghaffari, 2020]

Summary

- **MIS and Δ+1 coloring (randomized):** $O(\log n)$ [Luby '86]
- **Best deterministic algorithms:**
  - **MIS:** $O(\log^c n)$ rounds [Rozhoň, Ghaffari, 2019]
  - **Δ+1 coloring:** $O(\log^2 n \log \Delta)$ rounds [Kuhn, Ghaffari, 2020]
- **Best randomized algorithms:**

Summary

- **MIS and Δ+1 coloring (randomized):** **$O(\log n)$** [Luby '86]
- **Best deterministic algorithms:**
  - **MIS:** **$O(\log^c n)$ rounds** [Rozhoň, Ghaffari, 2019]
  - **Δ+1 coloring:** **$O(\log^2 n \log \Delta)$ rounds** [Kuhn, Ghaffari, 2020]
- **Best randomized algorithms:**
  - **MIS**: **$O(\log \Delta + \log^c \log n)$** [Ghaffari '16]

Summary

- **MIS and Δ+1 coloring (randomized):** **$O(\log n)$** [Luby '86]
- **Best deterministic algorithms:**
  - **MIS:** **$O(\log^c n)$ rounds** [Rozhoň, Ghaffari, 2019]
  - **Δ+1 coloring:** **$O(\log^2 n \log \Delta)$ rounds** [Kuhn, Ghaffari, 2020]
- **Best randomized algorithms:**
  - **MIS**: **$O(\log \Delta + \log^c \log n)$** [Ghaffari '16]
  - **Δ+1 coloring:** **$O(\log^c \log n)$** [Chang, Li, Pettie '18]

Summary

- **MIS and Δ+1 coloring (randomized):** $O(\log n)$ [Luby '86]
- **Best deterministic algorithms:**
  - **MIS:** $O(\log^c n)$ rounds [Rozhoň, Ghaffari, 2019]
  - **Δ+1 coloring:** $O(\log^2 n \log \Delta)$ rounds [Kuhn, Ghaffari, 2020]
- **Best randomized algorithms:**
  - **MIS**: $O(\log \Delta + \log^c \log n)$ [Ghaffari '16]
  - **Δ+1 coloring:** $O(\log^c \log n)$ [Chang, Li, Pettie '18]
- **Best lower bounds:**

Summary

- **MIS and $\Delta+1$ coloring (randomized):** $O(\log n)$ [Luby '86]
- **Best deterministic algorithms:**
    - **MIS:** $O(\log^c n)$ rounds [Rozhoň, Ghaffari, 2019]
    - **$\Delta+1$ coloring:** $O(\log^2 n \log \Delta)$ rounds [Kuhn, Ghaffari, 2020]
- **Best randomized algorithms:**
    - **MIS:** $O(\log \Delta + \log^c \log n)$ [Ghaffari '16]
    - **$\Delta+1$ coloring:** $O(\log^c \log n)$ [Chang, Li, Pettie '18]
- **Best lower bounds:**
    - **$\Delta+1$ coloring:** $\Omega(\log^* n)$ [Linial '87]

Summary

- **MIS and Δ+1 coloring (randomized):** $O(\log n)$ [Luby '86]
- **Best deterministic algorithms:**
  - **MIS:** $O(\log^c n)$ rounds [Rozhoň, Ghaffari, 2019]
  - **Δ+1 coloring:** $O(\log^2 n \log \Delta)$ rounds [Kuhn, Ghaffari, 2020]
- **Best randomized algorithms:**
  - **MIS**: $O(\log \Delta + \log^c \log n)$ [Ghaffari '16]
  - **Δ+1 coloring:** $O(\log^c \log n)$ [Chang, Li, Pettie '18]
- **Best lower bounds:**
  - **Δ+1 coloring:** $\Omega(\log^* n)$ [Linial '87]
  - **MIS (randomized):** $\Omega(\sqrt{(\log n / \log \log n)})$ [Kuhn, Moscibroda, Wattenhofer '04]

Summary

- **MIS and Δ+1 coloring (randomized):** $O(\log n)$ [Luby '86]
- **Best deterministic algorithms:**
  - **MIS:** $O(\log^c n)$ rounds [Rozhoň, Ghaffari, 2019]
  - **Δ+1 coloring:** $O(\log^2 n \log \Delta)$ rounds [Kuhn, Ghaffari, 2020]
- **Best randomized algorithms:**
  - **MIS**: $O(\log \Delta + \log^c \log n)$ [Ghaffari '16]
  - **Δ+1 coloring:** $O(\log^c \log n)$ [Chang, Li, Pettie '18]
- **Best lower bounds:**
  - **Δ+1 coloring:** $\Omega(\log^* n)$ [Linial '87]
  - **MIS (randomized):** $\Omega(\sqrt{(\log n / \log\log n)})$ [Kuhn, Moscibroda, Wattenhofer '04]
  - **MIS (deterministic):** $\Omega(\log n / \log\log n)$ [Balliu, Brandt, Hirvonen, Olivetti, Rabie, Suomela '19]

Summary

- **MIS and Δ+1 coloring (randomized):** $O(\log n)$ [Luby '86]
- **Best deterministic algorithms:**
  - **MIS:** $O(\log^c n)$ rounds [Rozhoň, Ghaffari, 2019]
  - **Δ+1 coloring:** $O(\log^2 n \log \Delta)$ rounds [Kuhn, Ghaffari, 2020]
- **Best randomized algorithms:**
  - **MIS:** $O(\log \Delta + \log^c \log n)$ [Ghaffari '16]
  - **Δ+1 coloring:** $O(\log^c \log n)$ [Chang, Li, Pettie '18]
- **Best lower bounds:**
  - **Δ+1 coloring:** $\Omega(\log^* n)$ [Linial '87]
  - **MIS (randomized):** $\Omega(\sqrt{(\log n / \log \log n)})$ [Kuhn, Moscibroda, Wattenhofer '04]
  - **MIS (deterministic):** $\Omega(\log n / \log \log n)$ [Balliu, Brandt, Hirvonen, Olivetti, Rabie, Suomela '19]
  - **MIS (deterministic, on trees):**

Summary

- **MIS and Δ+1 coloring (randomized):** $O(\log n)$ [Luby '86]
- **Best deterministic algorithms:**
  - **MIS:** $O(\log^c n)$ rounds [Rozhoň, Ghaffari, 2019]
  - **Δ+1 coloring:** $O(\log^2 n \log \Delta)$ rounds [Kuhn, Ghaffari, 2020]
- **Best randomized algorithms:**
  - **MIS:** $O(\log \Delta + \log^c \log n)$ [Ghaffari '16]
  - **Δ+1 coloring:** $O(\log^c \log n)$ [Chang, Li, Pettie '18]
- **Best lower bounds:**
  - **Δ+1 coloring:** $\Omega(\log^* n)$ [Linial '87]
  - **MIS (randomized):** $\Omega(\sqrt{(\log n / \log\log n)})$ [Kuhn, Moscibroda, Wattenhofer '04]
  - **MIS (deterministic):** $\Omega(\log n / \log\log n)$ [Balliu, Brandt, Hirvonen, Olivetti, Rabie, Suomela '19]
  - **MIS (deterministic, on trees):**
    - $O(\log n / \log\log n)$ [Barenboim, Elkin '08]

Summary

- **MIS and Δ+1 coloring (randomized):** $O(\log n)$ [Luby '86]
- **Best deterministic algorithms:**
  - **MIS:** $O(\log^c n)$ rounds [Rozhoň, Ghaffari, 2019]
  - **Δ+1 coloring:** $O(\log^2 n \log \Delta)$ rounds [Kuhn, Ghaffari, 2020]
- **Best randomized algorithms:**
  - **MIS:** $O(\log \Delta + \log^c \log n)$ [Ghaffari '16]
  - **Δ+1 coloring:** $O(\log^c \log n)$ [Chang, Li, Pettie '18]
- **Best lower bounds:**
  - **Δ+1 coloring:** $\Omega(\log^* n)$ [Linial '87]
  - **MIS (randomized):** $\Omega(\sqrt{\log n / \log \log n})$ [Kuhn, Moscibroda, Wattenhofer '04]
  - **MIS (deterministic):** $\Omega(\log n / \log \log n)$ [Balliu, Brandt, Hirvonen, Olivetti, Rabie, Suomela '19]
  - **MIS (deterministic, on trees):**
    - $O(\log n / \log \log n)$ [Barenboim, Elkin '08]
    - $\Omega(\log n / \log \log n)$ [Balliu, Brandt, Kuhn, Olivetti '21] <- **result from 3 weeks ago!**