Maximal Independent Set

Parallel Graph Algorithms

Outline

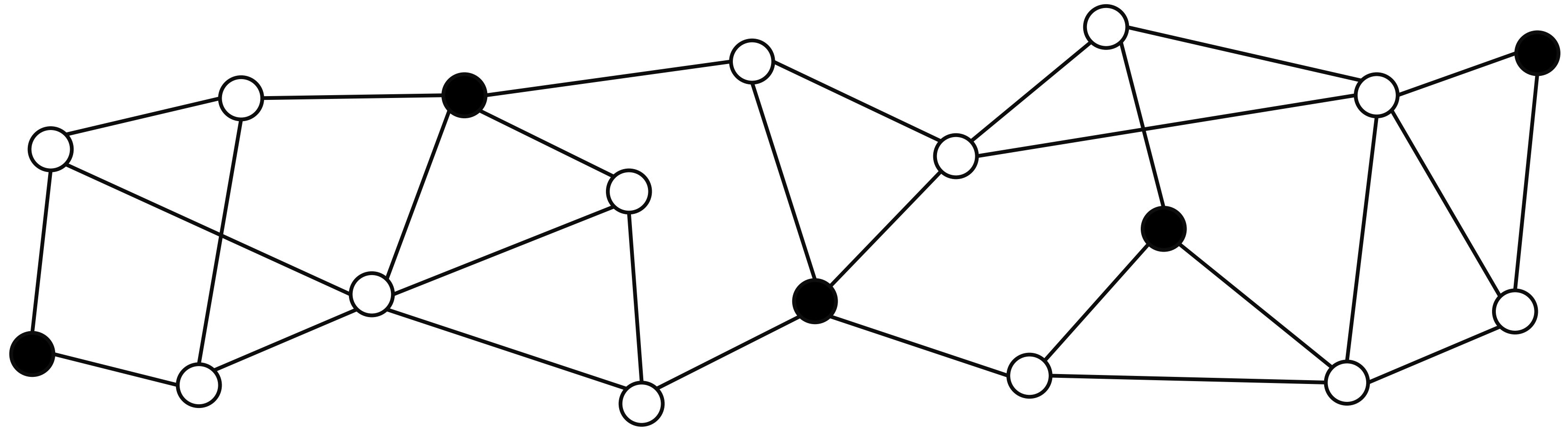
- Maximal Independent Set (MIS)
  - What is it?
  - Why are we interested in it?
- Bad Ideas
  - Greedy
  - “Local” greedy
- Luby’s Algorithm
- Literature

Outline

- Maximal Independent Set (MIS)
  - What is it?
  - Why are we interested in it?
- Bad Ideas
  - Greedy
  - “Local” greedy
- Luby’s Algorithm
- Literature

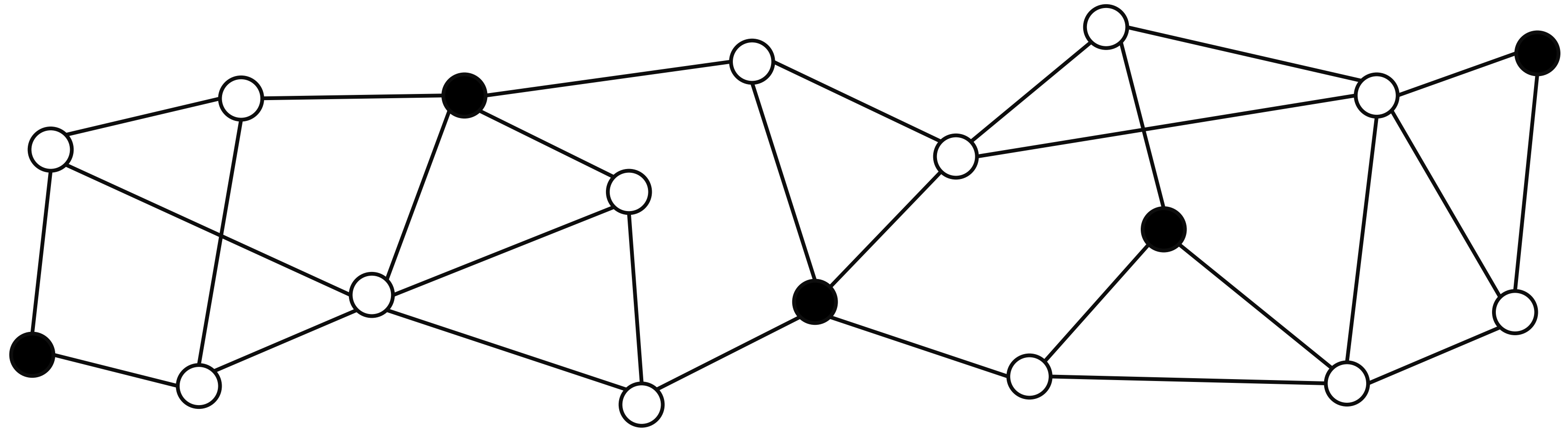
Learning objectives:
You are able to
- explain why finding a maximum independent set in the LOCAL model is hard
- state the MIS algorithm by Luby
- Analyse the probability that a neighbor of a good node gets selected to the MIS in one iteration of Luby

Maximal Independent Set (MIS)



Independent Set:
A set $I \subseteq V$ is *independent* if there are no edges between nodes in I.

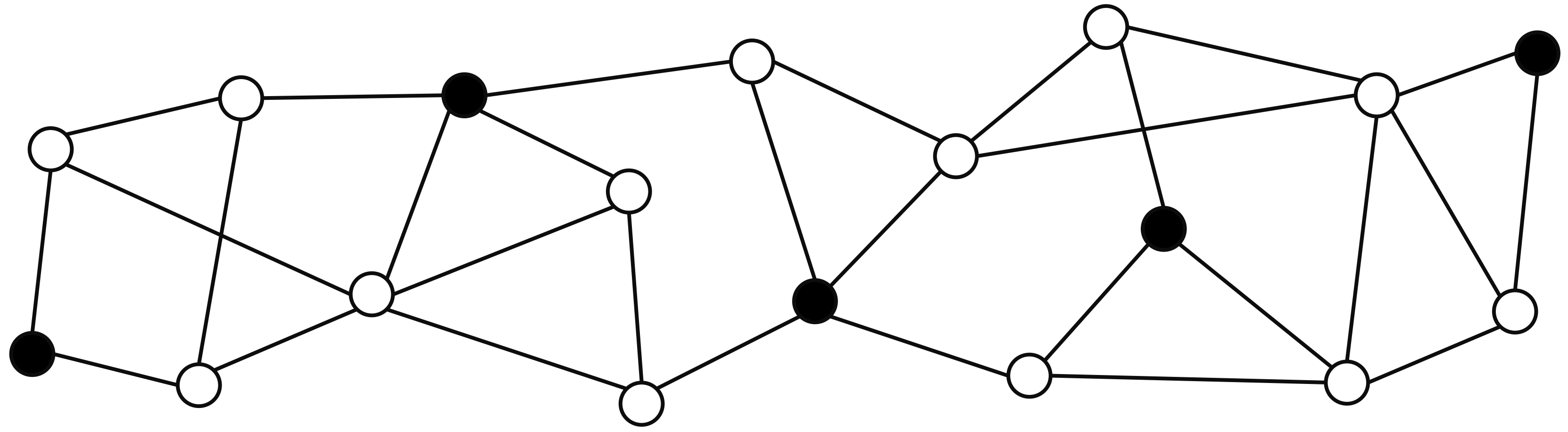
Maximal Independent Set (MIS)



Maximality:
An independent set $I \subseteq V$ is *maximal* if for any node $u \in V \setminus I$, it holds that $I \cup \{u\}$ is not independent.

Independent Set:
A set $I \subseteq V$ is *independent* if there are no edges between nodes in I.

Maximal Independent Set (MIS)



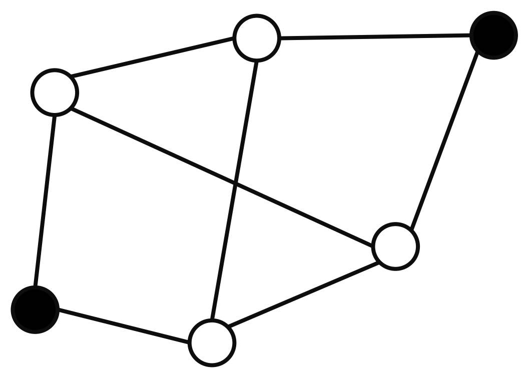
Maximality:
An independent set $I \subseteq V$ is *maximal* if for any node $u \in V \setminus I$, it holds that $I \cup \{u\}$ is not independent.

Independent Set:
A set $I \subseteq V$ is *independent* if there are no edges between nodes in I.

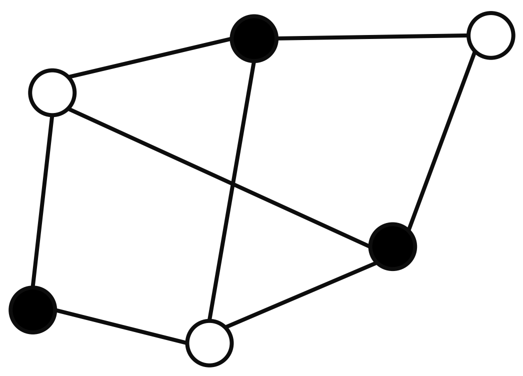
You cannot add any nodes to an MIS without breaking independence.

Why not Maximum?

Notice:
A maximal independent set does not need to be of maximum size.



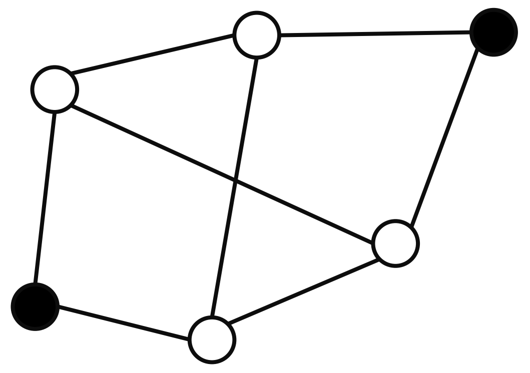
Maximal

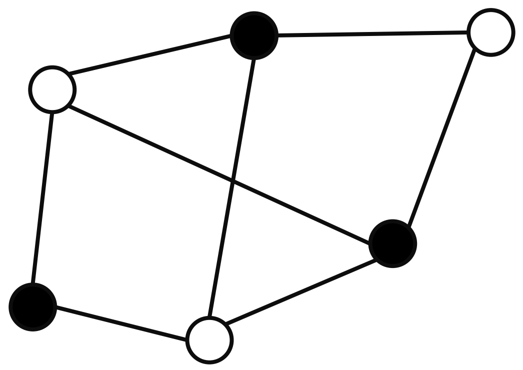


Maximum

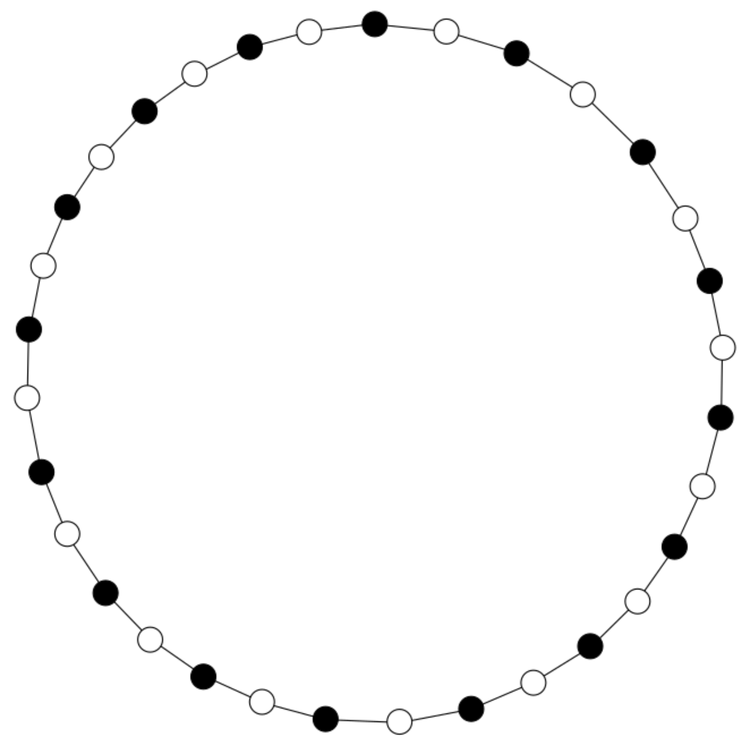
Why not Maximum?

Notice:
A maximal independent set does not need to be of maximum size.


Maximal

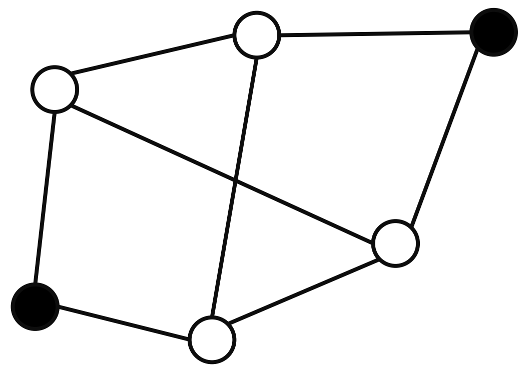

Maximum

Observation:
A maximum independent set on an even ring corresponds to a 2-coloring.

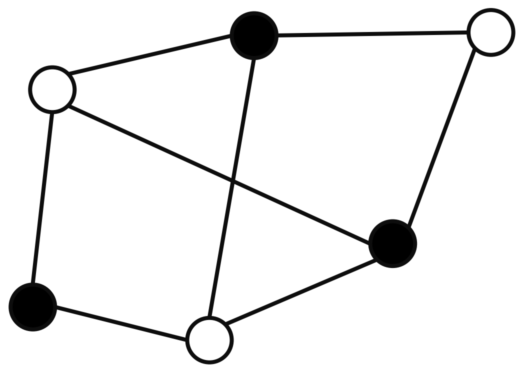


Why not Maximum?

Notice:
A maximal independent set does not need to be of maximum size.

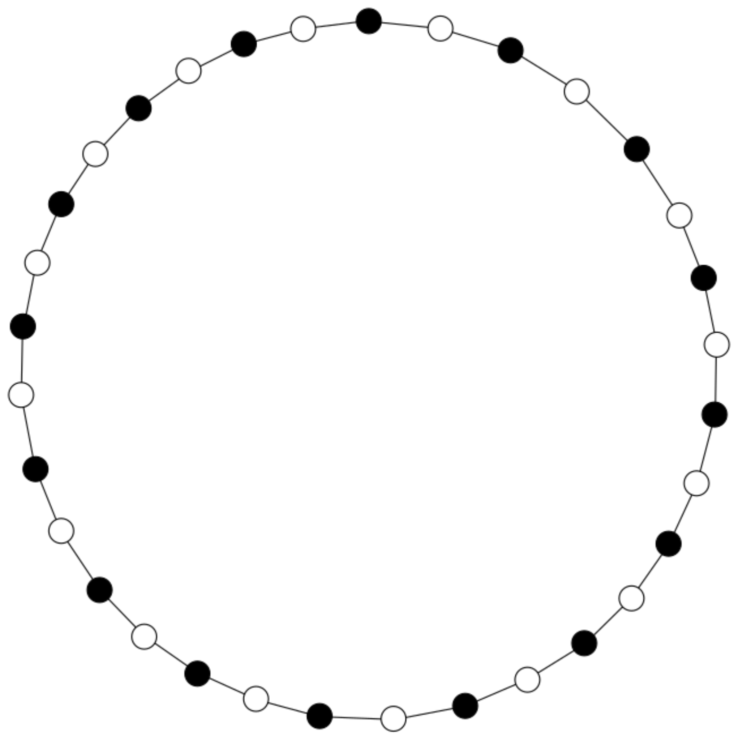


Maximal



Maximum

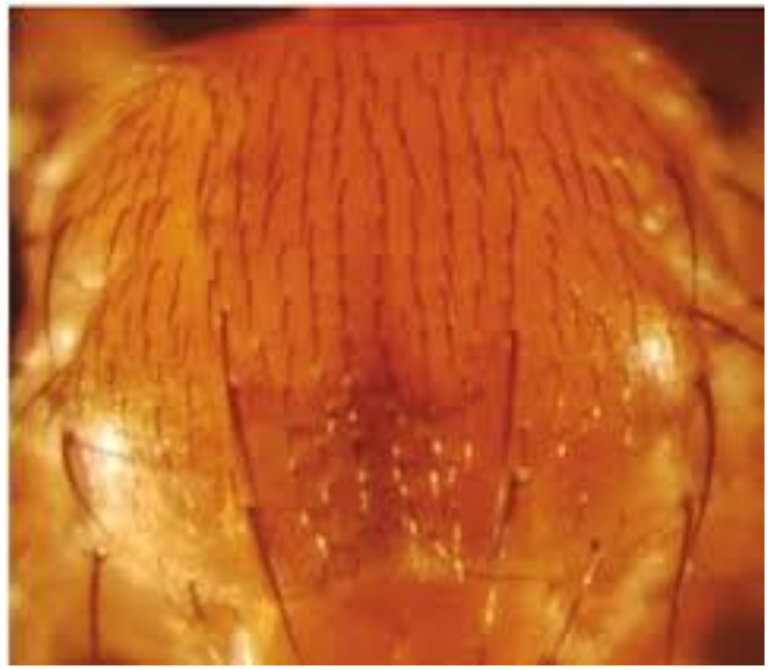
Observation:
A maximum independent set on an even ring corresponds to a 2-coloring.



Boring ☹

Why Should I Care?

Primitive symmetry breaking

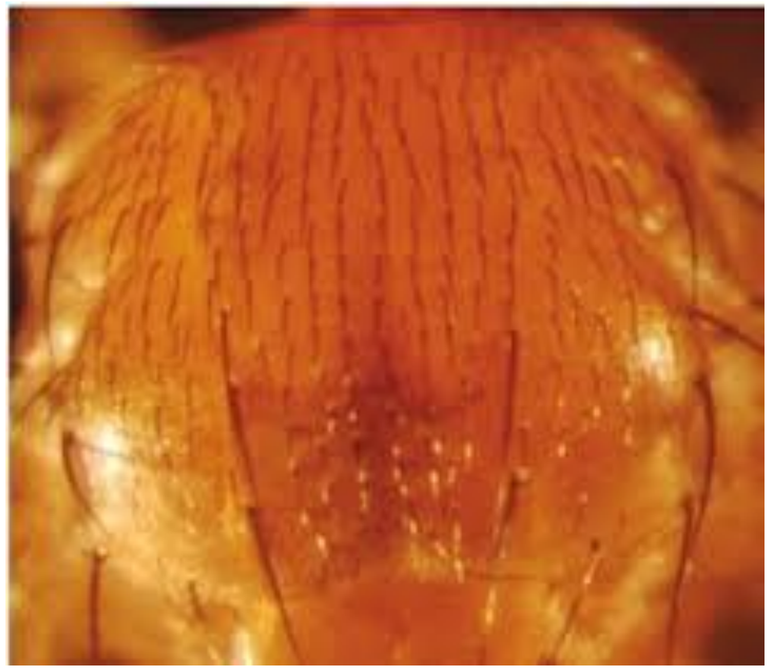


[Afek, Alon, Barad, Hornstein, Barkai, Bar-Joseph, Science 2011]

Why Should I Care?

Primitive symmetry breaking

It's not easy! Understand this first and then go for harder problems.



[Afek, Alon, Barad, Hornstein, Barkai, Bar-Joseph, Science 2011]

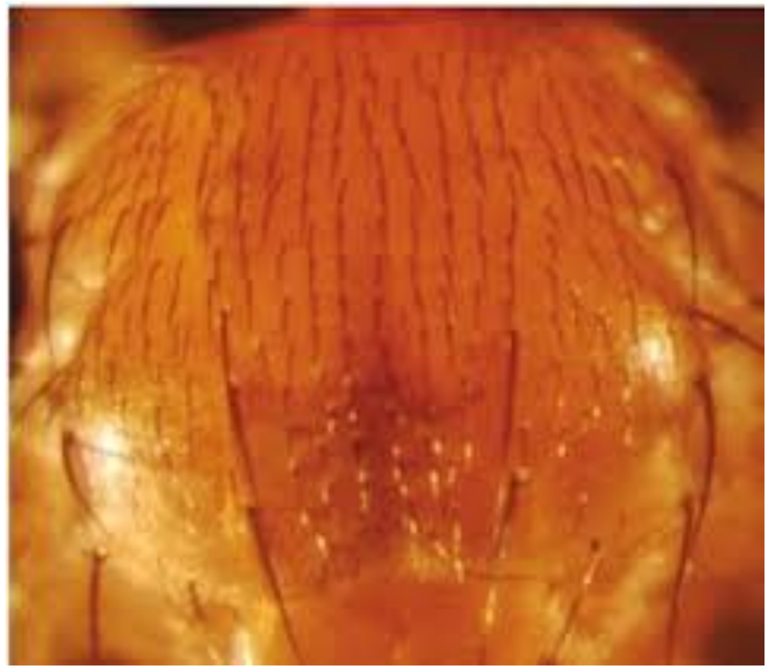
Why Should I Care?

Primitive symmetry breaking

It’s not easy! Understand this first and then go for harder problems.

Subroutine for more sophisticated tools such as distributed Lovász Local Lemma (LLL).

[Moser, Tardos, JACM 2010]



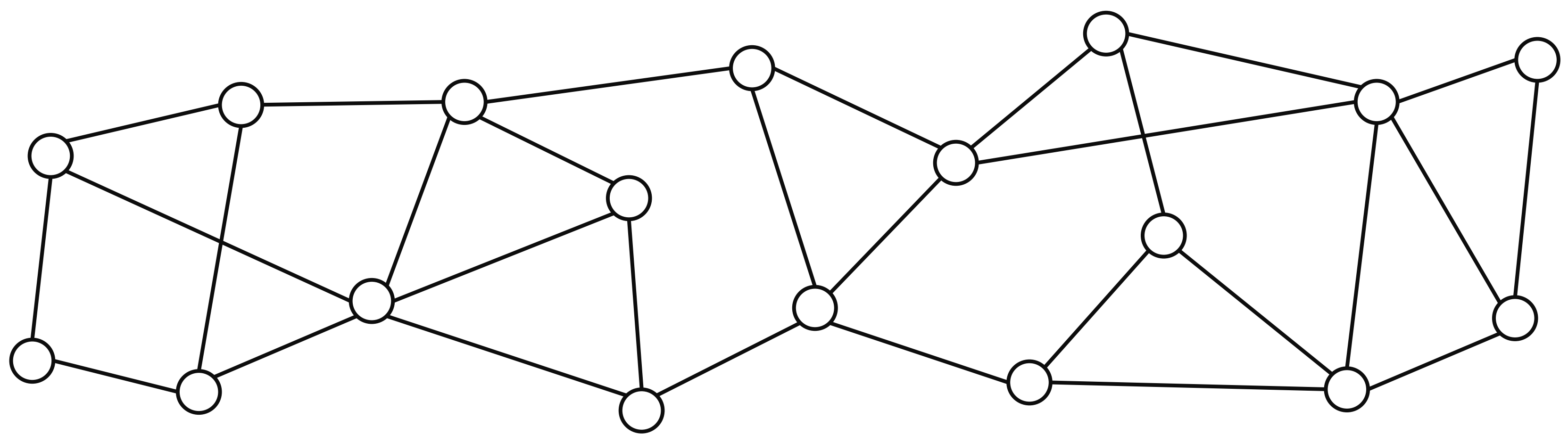
[Afek, Alon, Barad, Hornstein, Barkai, Bar-Joseph, Science 2011]

Outline

- Maximal Independent Set (MIS)
  - What is it?
  - Why are we interested in it?
- Bad Ideas
  - Greedy
  - “Local” greedy
- Luby’s Algorithm
- Literature

Greedy MIS

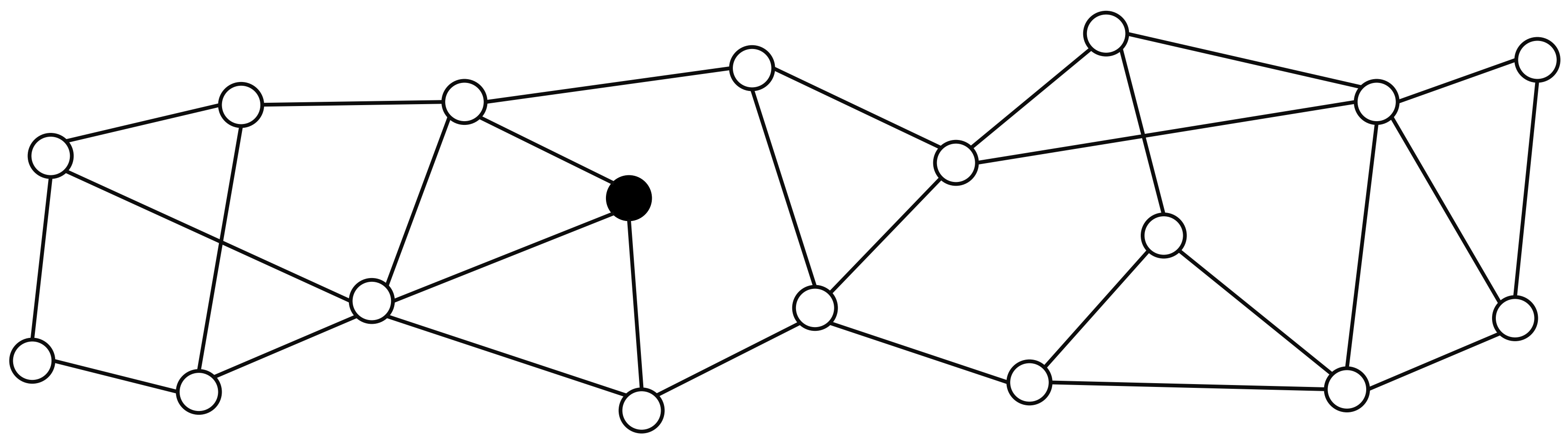
Greedy:
One by one, pick nodes into the MIS and remove neighbors from the graph.



What do we need to solve?

Just use the greedy algorithm…?

Greedy MIS



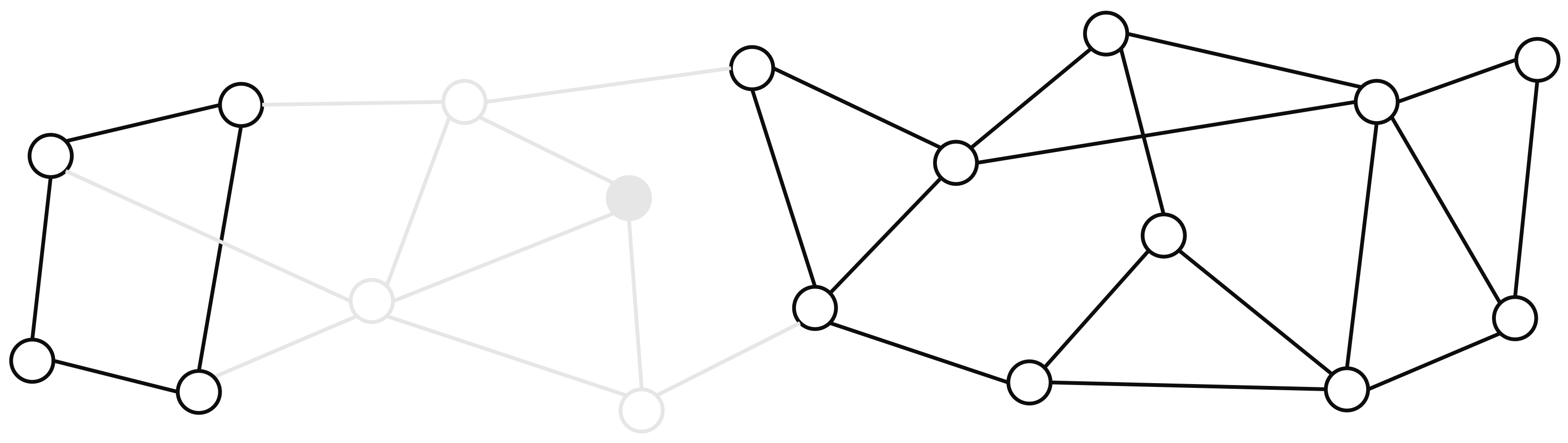
Greedy:
One by one, pick nodes into the MIS and remove neighbors from the graph.

What do we need to solve?

Just use the greedy algorithm…?

Greedy MIS

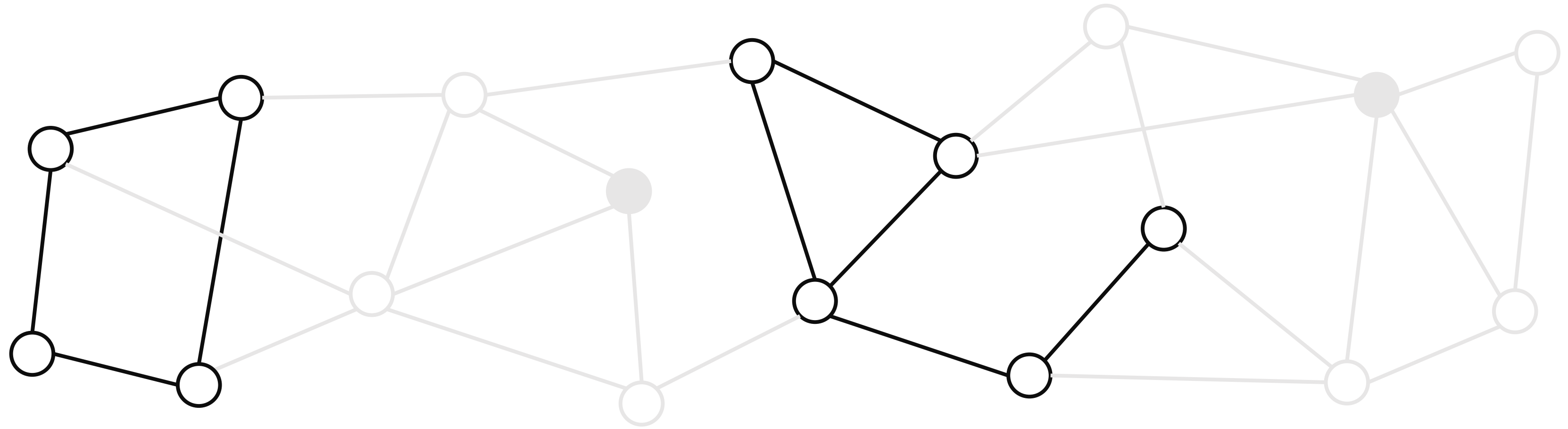
Greedy:
One by one, pick nodes into the MIS and remove neighbors from the graph.



What do we need to solve?

Just use the greedy algorithm…?

Greedy MIS

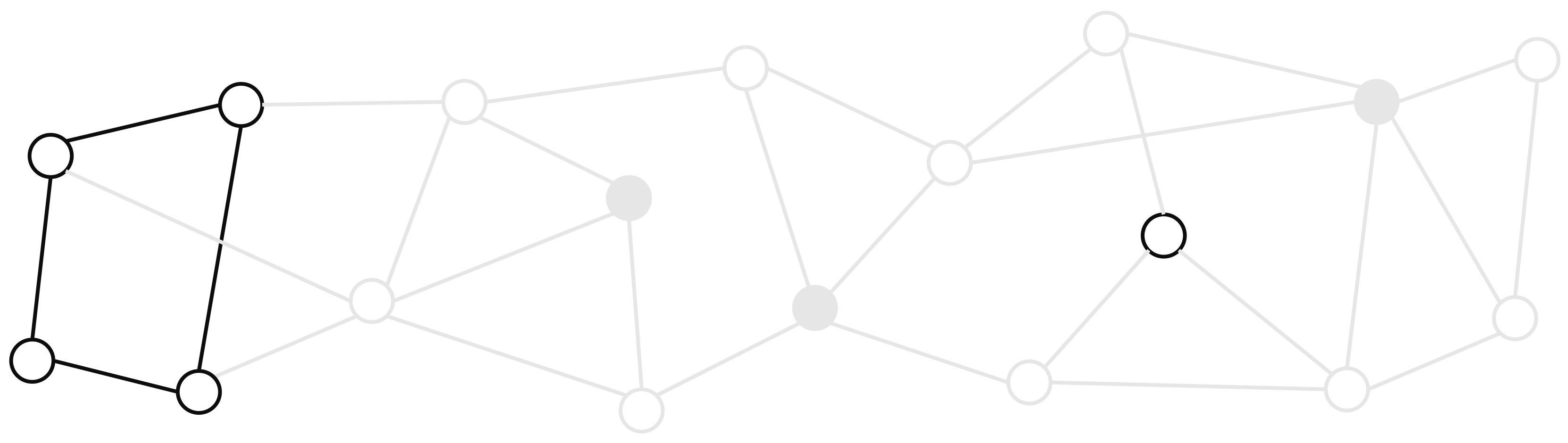


Greedy:
One by one, pick nodes into the MIS and remove neighbors from the graph.

What do we need to solve?

Just use the greedy algorithm…?

Greedy MIS



Greedy:
One by one, pick nodes into the MIS and remove neighbors from the graph.

What do we need to solve?

Just use the greedy algorithm…?

Greedy MIS

Greedy:
One by one, pick nodes into the MIS and remove neighbors from the graph.

Runtime: $\Theta(n)$

What do we need to solve?

Just use the greedy algorithm…?

Greedy MIS

Greedy:
One by one, pick nodes into the MIS and remove neighbors from the graph.

What do we need to solve?

Just use the greedy algorithm…?

Runtime: $\Theta(n)$

Not better than the trivial algorithm.

Greedy MIS

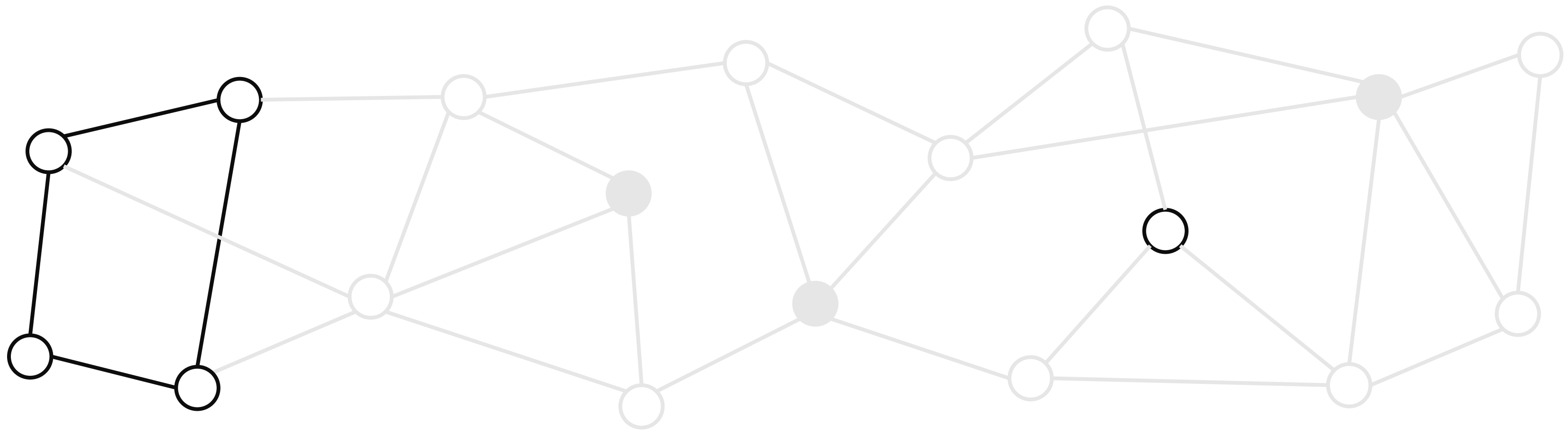
~~**Greedy:**~~
~~One by one, pick nodes into the MIS and remove neighbors from the graph.~~

Runtime: $\Theta(n)$

Greedy MIS

~~**Greedy:**
One by one, pick nodes into the MIS and remove neighbors from the graph.~~

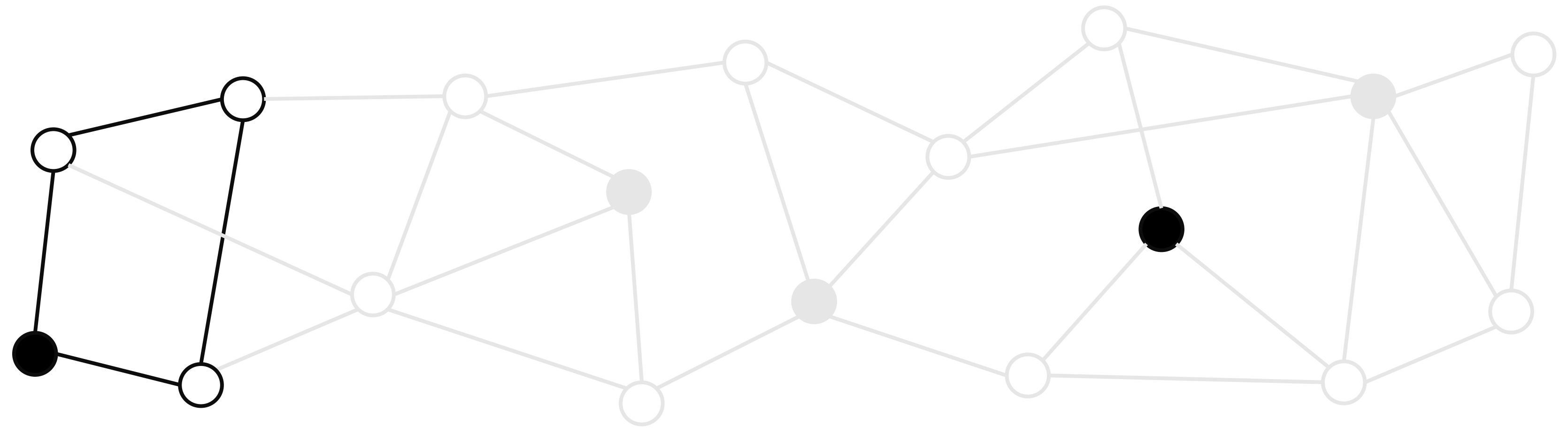
Runtime: $\Theta(n)$



Greedy MIS

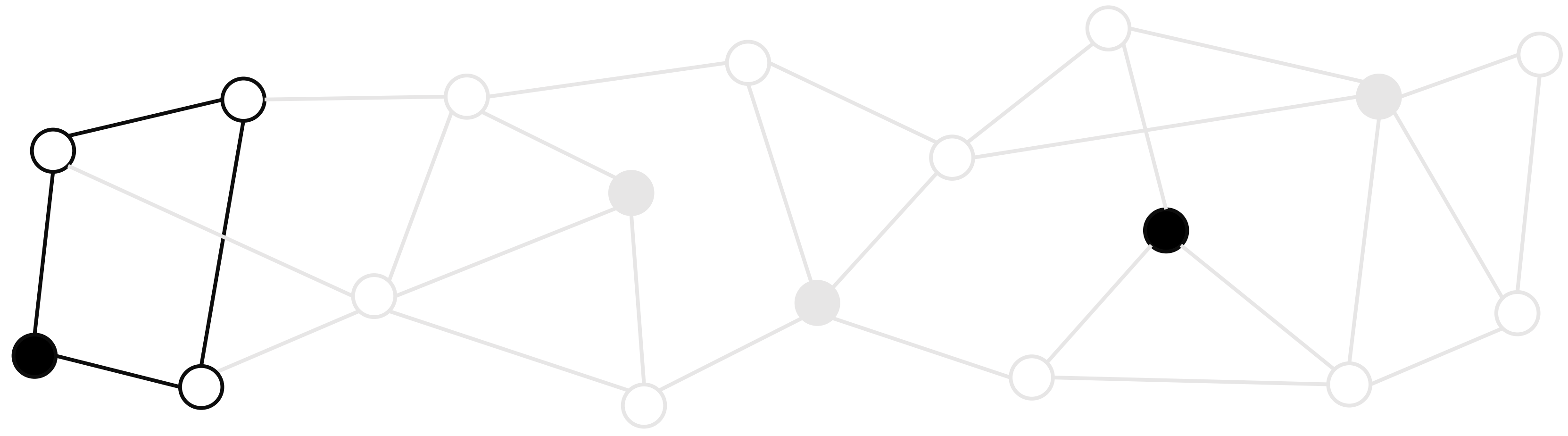
~~**Greedy:**~~
~~One by one, pick nodes into the MIS and remove neighbors from the graph.~~

Runtime: $\Theta(n)$



Could deal with both remaining components at once

Greedy MIS

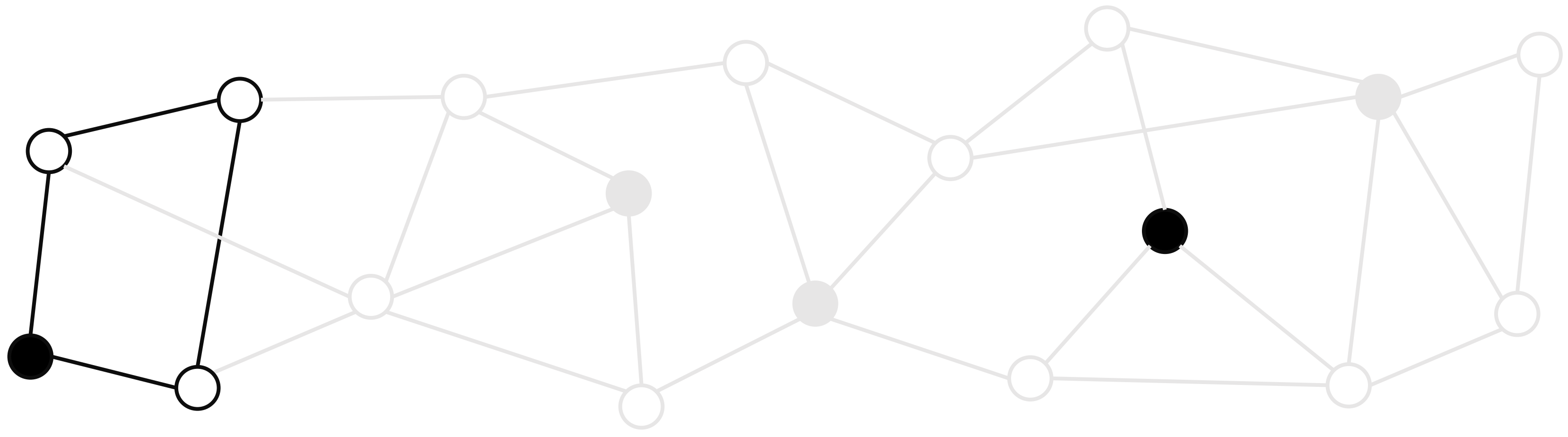


~~**Greedy:**
One by one, pick nodes into the MIS and remove neighbors from the graph.~~

Runtime: $\Theta(n)$

Choose local maxima, according to IDs?

Greedy MIS



~~**Greedy:**~~
~~One by one, pick nodes into the MIS and remove neighbors from the graph.~~

Runtime: $\Theta(n)$

Choose local maxima, according to IDs?

Think about a cycle with monotonically increasing IDs. One local minimum and maximum at once.

☹

Outline

- Maximal Independent Set (MIS)
  - What is it?
  - Why are we interested in it?
- Bad Ideas
  - Greedy
  - “Local” greedy
- Luby’s Algorithm
- Literature

Luby's Algorithm

[Luby J. Computing 86]
[Alon, Babai, Itai J. Algorithms 86]

Very informally:
Randomly pick the local maxima.

Algorithm (one phase)
Each Node u:
With probability $\frac{1}{2d(u)}$ mark u.

If u is marked and no node $v \in N(u)$ with $d(v) \geq d(u)$ is marked:
Select u to the MIS and remove u and all $N(u)$ from the graph.

In the end of the phase, unmark every node.

Luby's Algorithm

Change over time

Algorithm (one phase)

Each Node u:

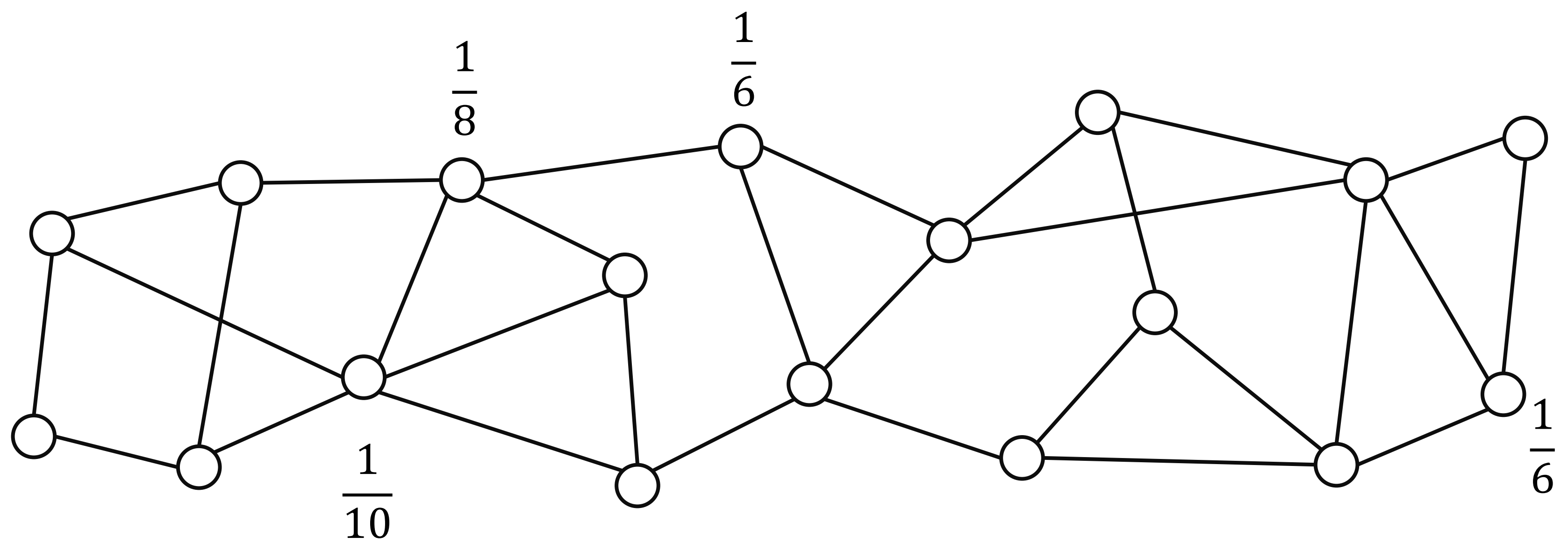
With probability $\frac{1}{2d(u)}$ mark u.

If u is marked and no node $v \in N(u)$ with $d(v) \geq d(u)$ is marked:

Select u to the MIS and remove u and all $N(u)$ from the graph.

In the end of the phase, unmark every node.

Luby's Algorithm



Algorithm (one phase)

Each Node u:

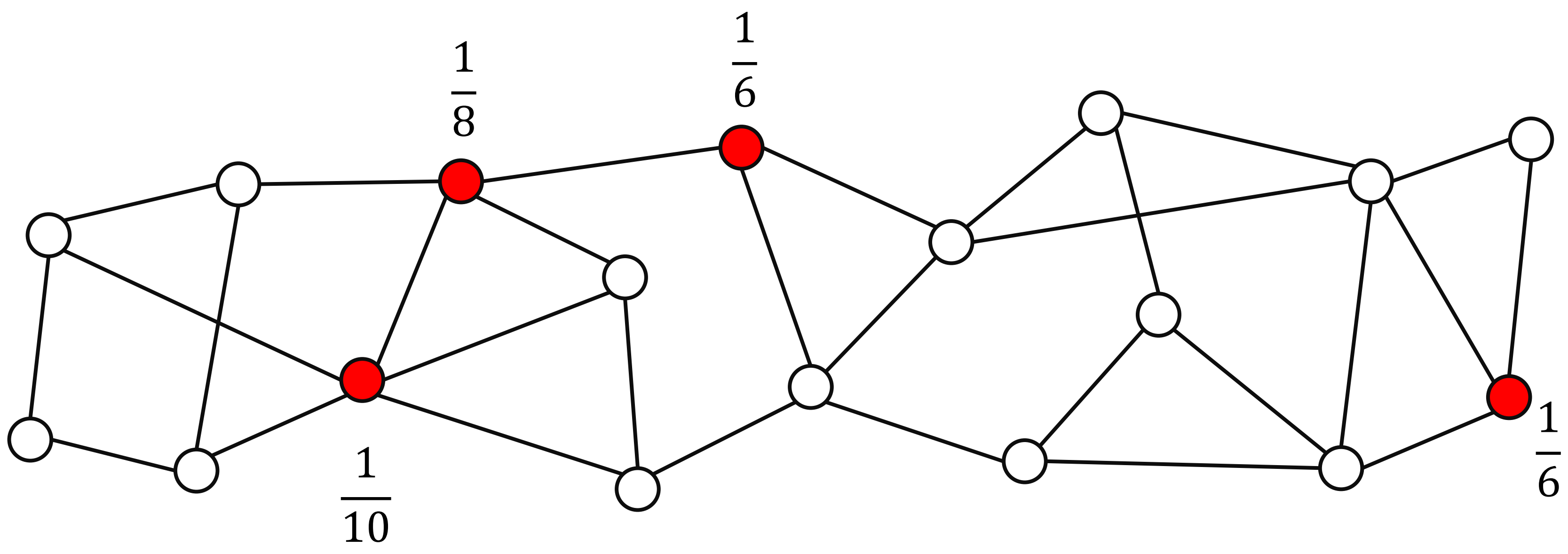
With probability $\frac{1}{2d(u)}$ mark u.

If u is marked and no node $v \in N(u)$ with $d(v) \geq d(u)$ is marked:

Select u to the MIS and remove u and all $N(u)$ from the graph.

In the end of the phase, unmark every node.

Luby’s Algorithm



Algorithm (one phase)

Each Node u:

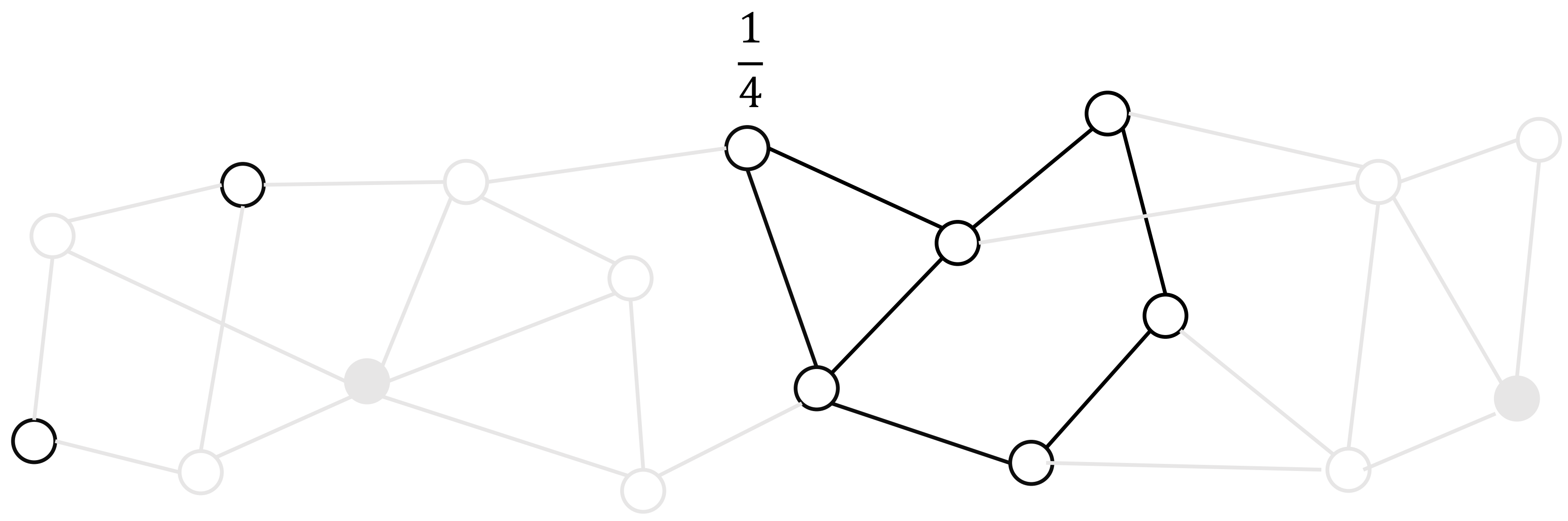
With probability $\frac{1}{2d(u)}$ mark u.

If u is marked and no node $v \in N(u)$ with $d(v) \geq d(u)$ is marked:

Select u to the MIS and remove u and all $N(u)$ from the graph.

In the end of the phase, unmark every node.

Luby’s Algorithm



Algorithm (one phase)

Each Node u:

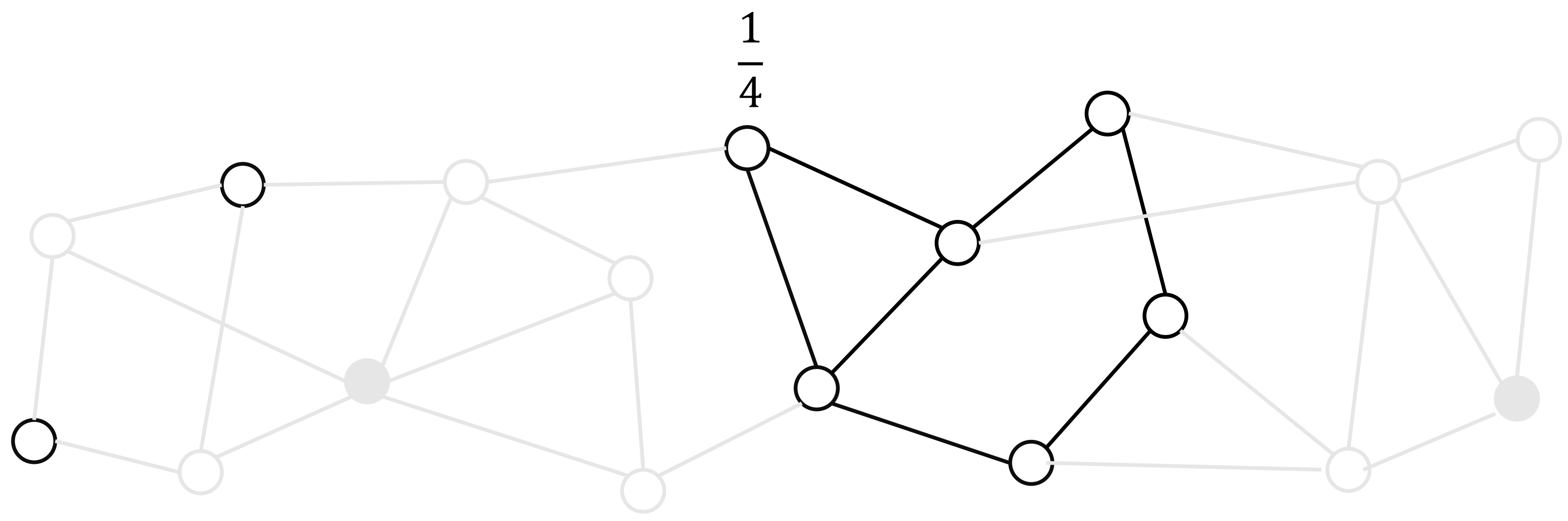
With probability $\frac{1}{2d(u)}$ mark u.

If u is marked and no node $v \in N(u)$ with $d(v) \geq d(u)$ is marked:

Select u to the MIS and remove u and all $N(u)$ from the graph.

In the end of the phase, unmark every (remaining) node.

Luby's Algorithm



Algorithm (one phase)

Each Node u:

With probability $\frac{1}{2d(u)}$ mark u.

If u is marked and no node $v \in N(u)$ with $d(v) \geq d(u)$ is marked:

Select u to the MIS and remove u and all $N(u)$ from the graph.

In the end of the phase, unmark every (remaining) node.

Luby's Algorithm

Challenge:
What kind of progress do we have?

Does a node get picked to the MIS with a constant probability?

Algorithm (one phase)
Each Node u:
With probability $\frac{1}{2d(u)}$ mark u.

If u is marked and no node $v \in N(u)$ with $d(v) \geq d(u)$ is marked:
Select u to the MIS and remove u and all $N(u)$ from the graph.

In the end of the phase, unmark every (remaining) node.

Luby's Algorithm

Challenge:
What kind of progress do we have?

Does a node get picked to the MIS with a constant probability? No.

Algorithm (one phase)
Each Node u:
With probability $\frac{1}{2d(u)}$ mark u.

If u is marked and no node $v \in N(u)$ with $d(v) \geq d(u)$ is marked:
Select u to the MIS and remove u and all $N(u)$ from the graph.

In the end of the phase, unmark every (remaining) node.

Luby's Algorithm

Challenge:
What kind of progress do we have?

Does a node get picked to the MIS with a constant probability?

Does a node get removed with a constant probability?

Algorithm (one phase)
Each Node u:
With probability $\frac{1}{2d(u)}$ mark u.

If u is marked and no node $v \in N(u)$ with $d(v) \geq d(u)$ is marked:
Select u to the MIS and remove u and all $N(u)$ from the graph.

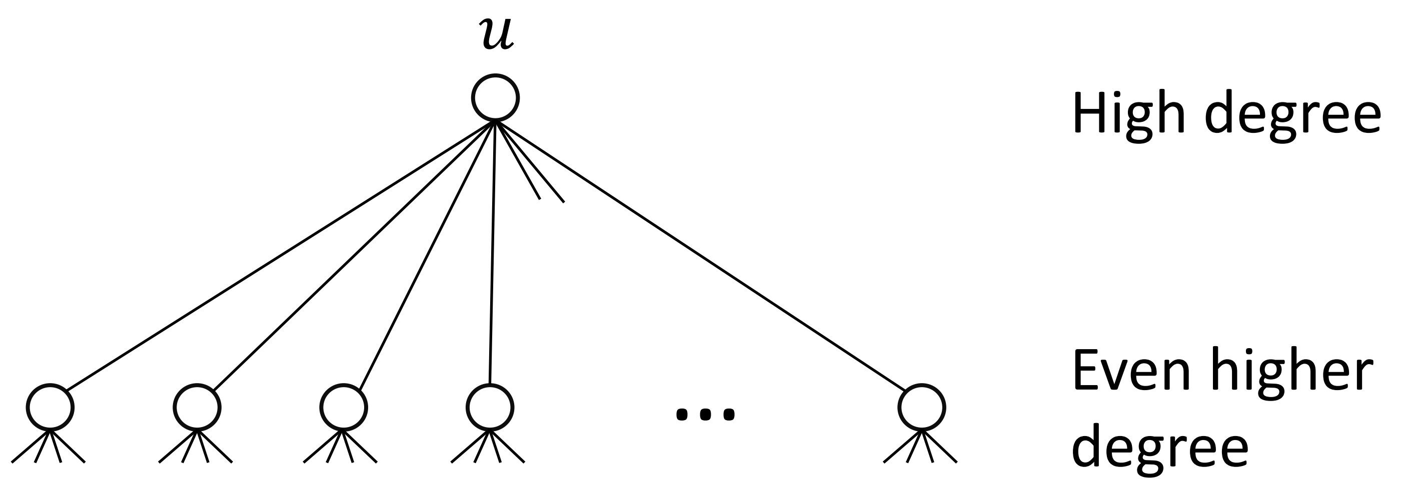
In the end of the phase, unmark every (remaining) node.

Luby's Algorithm

Challenge:
What kind of progress do we have?

Does a node get picked to the MIS with a constant probability?

Does a node get removed with a constant probability?

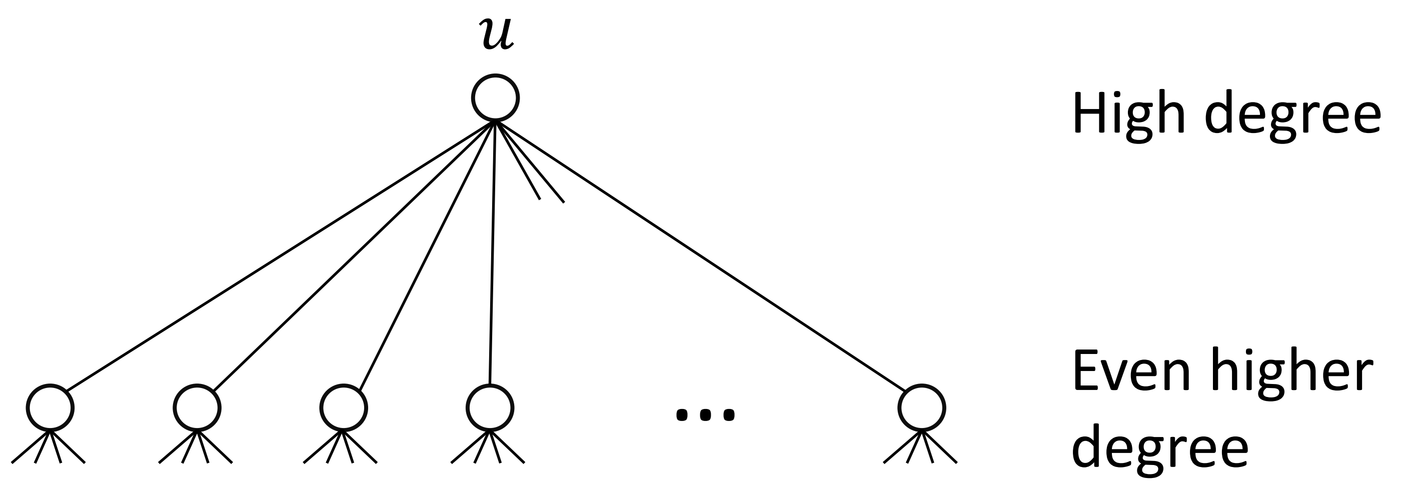


Luby’s Algorithm

Challenge:
What kind of progress do we have?

Does a node get picked to the MIS with a constant probability?

Does a node get removed with a constant probability?



$P[u$ gets selected$]$ is small

$P[$some $u \in N(u)$ gets selected$]$ is small.

☹

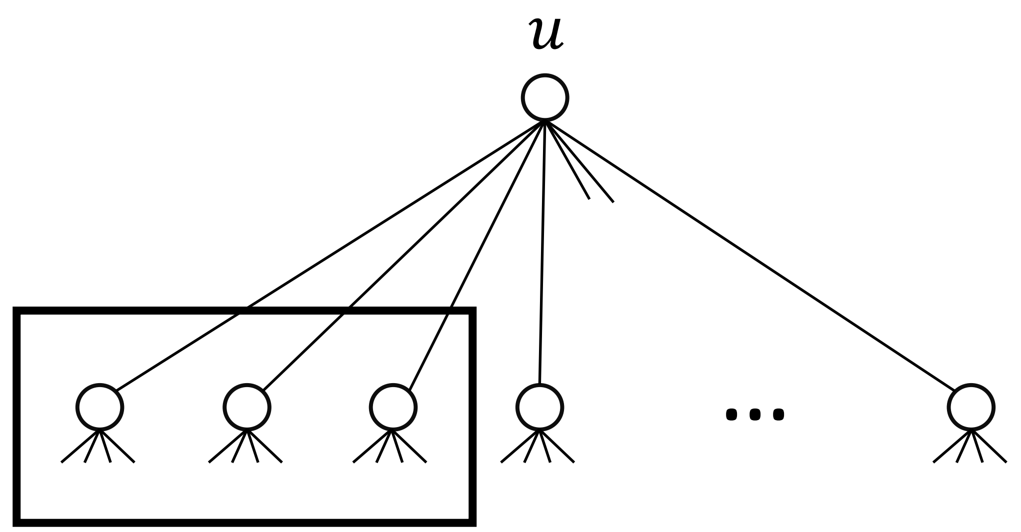
Luby's Algorithm

Node u is *good* if

$$|\{v \in N(u) \mid d(u) > d(v)\}| \geq \frac{d(u)}{3}$$

At least one third of the neighbors of u have smaller degree.

An edge is *good* if it is incident on a good node.



Intuition:
There is a reasonable chance that one is selected to the MIS.

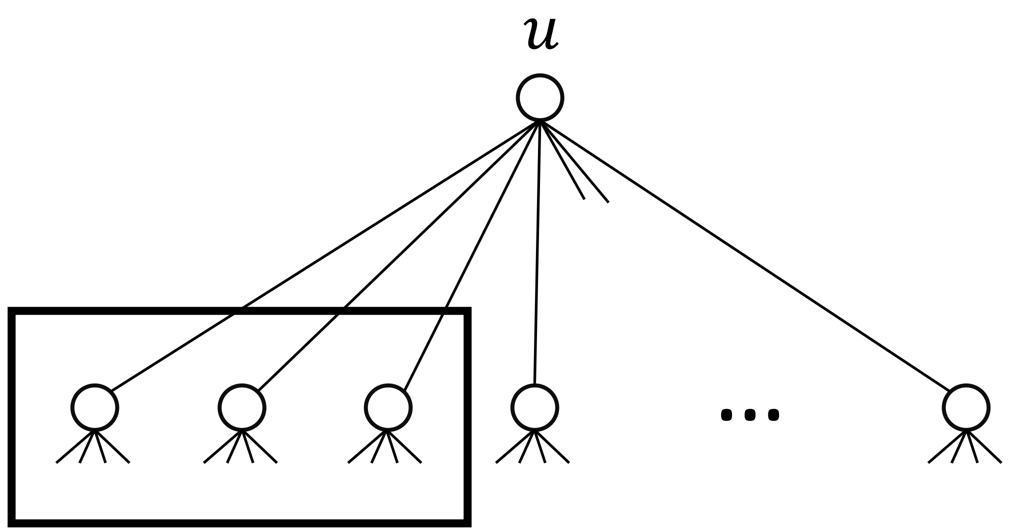
Luby's Algorithm

Node u is *good* if

$$|\{v \in N(u) \,|d(u) > d(v)\}| \geq \frac{d(u)}{3}$$

At least one third of the neighbors of u have smaller degree.

An edge is *good* if it is incident on a good node.



Lemma:
A good edge gets removed with a constant probability.

Lemma:
At least half of all edges are good.

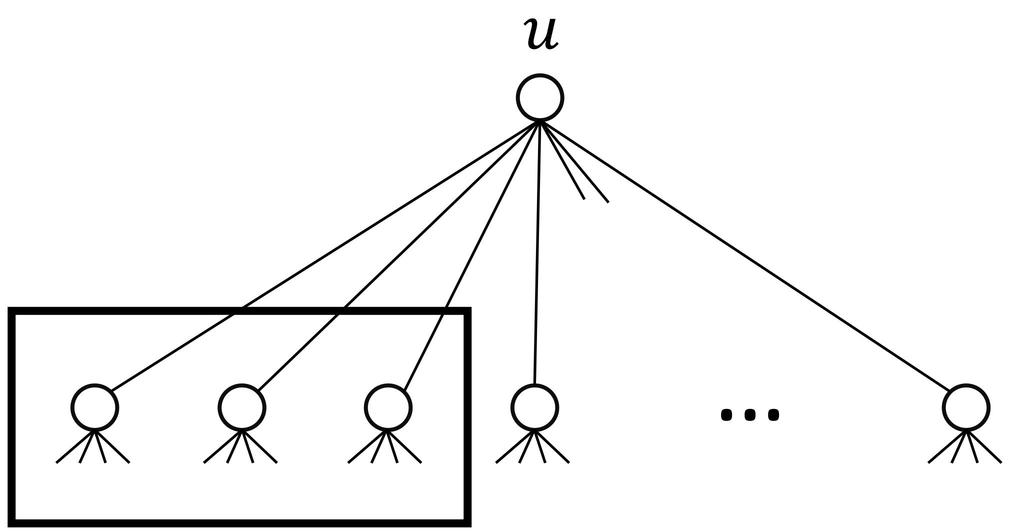
Luby's Algorithm

Node u is *good* if

$$|\{v \in N(u) \mid d(u) > d(v)\}| \geq \frac{d(u)}{3}$$

At least one third of the neighbors of u have smaller degree.

An edge is *good* if it is incident on a good node.



Lemma:
A good edge gets removed with a constant probability.

Lemma:
At least half of all edges are good.

Luby’s Algorithm

Lemma:
A good edge gets removed with a constant probability.

Lemma:
At least half of all edges are good.

Corollary:
After expected $O(\log m)$ rounds, no edges remain.
We can get this in expectation and w.h.p.

Luby's Algorithm

Lemma:
A good node gets removed with a constant probability.

Node u is *good* if

$$|\{v \in N(u) \mid d(u) > d(v)\}| \geq \frac{\deg u}{3}$$

What would be nice (Event R):
A "low degree" neighbor v of u is marked and no higher degree neighbor of v is marked.

Event $B(v)$:
v is marked and no higher degree neighbor of v is marked.

Luby’s Algorithm

Event $B(v)$:
v is marked and no higher degree neighbor of v is marked.

$$\mathrm{P}(B(v)) \geq \frac{1}{2d(v)}\left(1 - \frac{1}{2d(v)}\right)^{d(v)}$$

Luby's Algorithm

$$\left(1-\frac{x}{2}\right) \geq 2^{-x}$$

Event $B(v)$:
v is marked and no higher degree neighbor of v is marked.

$$\mathrm{P}(B(v)) \geq \frac{1}{2d(v)}\left(1-\frac{1}{2d(v)}\right)^{d(v)}$$

$$\geq \frac{1}{2d(v)} \cdot \left(2^{1/d(v)}\right)^{d(v)} = \frac{1}{4d(v)}$$

Luby's Algorithm

Event $B(v)$:
v is marked and no higher degree neighbor of v is marked.
$\mathrm{P}(B(v)) \geq \frac{1}{4d(v)}$

A neighbor v with $d(v) \leq 3$ is selected to the MIS with probability at least $1/12$.

Hence, we can assume that each $d(v) > 3$.

Lemma:
A good node gets removed with a constant probability.

Luby’s Algorithm

Node u is *good* if

$$|\{v \in N(u) \mid d(u) > d(v)\}| \geq \frac{d(u)}{3}$$

Let C be the lower degree neighbors of u.

Each $d(v) > 3$.

Luby's Algorithm

Node u is *good* if

$$|\{v \in N(u) \,|d(u) > d(v)\}| \geq \frac{d(u)}{3}$$

Let C be the lower degree neighbors of u.

Each $d(v) > 3$.

By definition:

$$\Sigma_{v \in C} \frac{1}{2d(v)} \geq \frac{d(u)}{3} \frac{1}{2d(v)}$$

Luby's Algorithm

Node u is *good* if

$$|\{v \in N(u) \,| d(u) > d(v)\}| \geq \frac{d(u)}{3}$$

Let C be the lower degree neighbors of u.

By definition:

$$\Sigma_{v \in C} \frac{1}{2d(v)} \geq \frac{d(u)}{3} \frac{1}{2d(v)}$$

$$\frac{d(u)}{3} \frac{1}{2d(u)} \geq \frac{1}{6}$$

Each $d(v) > 3$.

Luby's Algorithm

Node u is *good* if

$$|\{v \in N(u)\ | d(u) > d(v)\}| \geq \frac{d(u)}{3}$$

Let C be the lower degree neighbors of u.

By definition:

$$\Sigma_{v \in C} \frac{1}{2d(v)} \geq \frac{d(u)}{3} \frac{1}{2d(v)}$$

$$\frac{d(u)}{3} \frac{1}{2d(u)} \geq \frac{1}{6}$$

Each $d(v) > 3$.

There must exist $S \subseteq C$ such that

$$\frac{1}{6} \leq \Sigma_{v \in S} \frac{1}{2d(v)} \leq \frac{1}{3}$$

Luby's Algorithm

Inclusion-Exclusion:

Let S be a set such that $\frac{1}{6} \leq \Sigma_{v \in S} \frac{1}{2d(v)} \leq \frac{1}{3}$.

$P(R)$
$\geq \Sigma_{v \in S} P\big(B(v)\big) - \Sigma_{v \neq w \in S} P\big(B(v) \text{ and } B(w)\big)$

Event $B(v)$:
v is marked and no higher degree neighbor of v is marked.

$\mathrm{P}(B(v)) \geq \frac{1}{4d(v)}$

Event R:
A neighbor v of u is marked and no higher degree neighbor of v is marked.

Luby's Algorithm

Inclusion-Exclusion:

Let S be a set such that $\frac{1}{6} \leq \Sigma_{v \in S} \frac{1}{2d(v)} \leq \frac{1}{3}$.

$P(R)$

$\geq \Sigma_{v \in S} P\big(B(v)\big) - \Sigma_{v \neq w \in S} P\big(B(v) \text{ and } B(w)\big)$

$\geq \Sigma_{v \in S} \frac{1}{4d(v)} - \Sigma_{v \neq w \in S} P\big(B(v) \text{ and } B(w)\big)$

Event $B(v)$:

v is marked and no higher degree neighbor of v is marked.

$\mathrm{P}(B(v)) \geq \frac{1}{4d(v)}$

Event R:

A neighbor v of u is marked and no higher degree neighbor of v is marked.

Luby's Algorithm

Inclusion-Exclusion:

Let S be a set such that $\frac{1}{6} \leq \Sigma_{v \in S} \frac{1}{2d(v)} \leq \frac{1}{3}$.

$P(R)$

$\geq \Sigma_{v \in S} P\big(B(v)\big) - \Sigma_{v \neq w \in S} P\big(B(v) \text{ and } B(w)\big)$

$\geq \Sigma_{v \in S} \frac{1}{4d(v)} - \Sigma_{v \neq w \in S} \frac{1}{2d(v)} \frac{1}{2d(w)}$

Event $B(v)$:
v is marked and no higher degree neighbor of v is marked.

$\mathrm{P}(B(v)) \geq \frac{1}{4d(v)}$

Event R:
A neighbor v of u is marked and no higher degree neighbor of v is marked.

They both need to get marked at the least.

Luby's Algorithm

Inclusion-Exclusion:

Let S be a set such that $\frac{1}{6} \le \Sigma_{v\in S} \frac{1}{2d(v)} \le \frac{1}{3}$.

$P(R)$

$\ge \Sigma_{v\in S} P\big(B(v)\big) - \Sigma_{v\ne w\in S} P\big(B(v) \text{ and } B(w)\big)$

$$\ge \Sigma_{v\in S} \frac{1}{4d(v)} - \Sigma_{v\ne w\in S} \frac{1}{2d(v)} \frac{1}{2d(w)}$$

$$= \Sigma_{v\in S} \frac{1}{2d(v)} \left(\frac{1}{2} - \Sigma_{w\in S} \frac{1}{2d(w)}\right)$$

Event $B(v)$:
v is marked and no higher degree neighbor of v is marked.

$\mathrm{P}(B(v)) \ge \frac{1}{4d(v)}$

Event R:
A neighbor v of u is marked and no higher degree neighbor of v is marked.

Luby's Algorithm

Inclusion-Exclusion:

Let S be a set such that $\frac{1}{6} \leq \Sigma_{v\in S} \frac{1}{2d(v)} \leq \frac{1}{3}$.

$P(R)$

$\geq \Sigma_{v\in S} P\big(B(v)\big) - \Sigma_{v\neq w\in S} P\big(B(v) \text{ and } B(w)\big)$

$\geq \Sigma_{v\in S} \dfrac{1}{4d(v)} - \Sigma_{v\neq w\in S} \dfrac{1}{2d(v)} \dfrac{1}{2d(w)}$

$= \Sigma_{v\in S} \dfrac{1}{2d(v)} \left(\dfrac{1}{2} - \Sigma_{w\in S} \dfrac{1}{2d(w)}\right)$

$\geq \Sigma_{v\in S} \dfrac{1}{2d(v)} \left(\dfrac{1}{2} - \dfrac{1}{3}\right) \geq \dfrac{1}{6}\left(\dfrac{1}{6}\right) = \dfrac{1}{36}$

Event $B(v)$:

v is marked and no higher degree neighbor of v is marked.

$\mathrm{P}(B(v)) \geq \frac{1}{4d(v)}$

Event R:

A neighbor v of u is marked and no higher degree neighbor of v is marked.

Luby's Algorithm

Inclusion-Exclusion:

Let S be a set such that $\frac{1}{6} \le \Sigma_{v \in S} \frac{1}{2d(v)} \le \frac{1}{3}$.

$$P(R)$$
$$\ge \Sigma_{v \in S} P\big(B(v)\big) - \Sigma_{v \neq w \in S} P\big(B(v) \text{ and } B(w)\big)$$
$$\ge \Sigma_{v \in S} \frac{1}{4d(v)} - \Sigma_{v \neq w \in S} \frac{1}{2d(v)} \frac{1}{2d(w)}$$
$$= \Sigma_{v \in S} \frac{1}{2d(v)} \left(\frac{1}{2} - \Sigma_{w \in S} \frac{1}{2d(w)}\right)$$
$$\ge \Sigma_{v \in S} \frac{1}{2d(v)} \left(\frac{1}{2} - \frac{1}{3}\right) \ge \frac{1}{6}\left(\frac{1}{6}\right) = \frac{1}{36}$$

Event $B(v)$:
v is marked and no higher degree neighbor of v is marked.

$\mathrm{P}(B(v)) \ge \frac{1}{4d(v)}$

Event R:
A neighbor v of u is marked and no higher degree neighbor of v is marked.

Luby's Algorithm

Definition:
Each good edge is adjacent to a good node u.

Event R:
A neighbor v of u is marked and no higher degree neighbor of v is marked.

$$P(R) \geq \frac{1}{36}$$

Luby’s Algorithm

Lemma:
A good edge gets removed with a constant probability.

Lemma:
At least half of all edges are good.

Corollary:
After expected $O(\log m)$ rounds, no edges remain.
We can get this in expectation and w.h.p.

Luby's Algorithm

Lemma:
At least half of all edges are good.

Approach:
Find an upper bound on the number of bad edges.

Luby’s Algorithm

Lemma:
At least half of all edges are good.

Approach:
Find an upper bound on the number of bad edges.

Smart counting:
Orient edges towards higher degree neighbors.

Luby’s Algorithm

Smart counting:
Orient edges towards higher degree neighbors.

Lemma:
At least half of all edges are good.

Node u is *good* if

$$|\{v \in N(u) \mid d(u) > d(v)\}| \geq \frac{d(u)}{3}$$

Approach:
Find an upper bound on the number of bad edges.

Luby's Algorithm

Lemma:
At least half of all edges are good.

Approach:
Find an upper bound on the number of bad edges.

Smart counting:
Orient edges towards higher degree neighbors.

Node u is *good* if

$$|\{v \in N(u) \mid d(u) > d(v)\}| \geq \frac{d(u)}{3}$$

A bad node u has outdegree of at least $2d(u)/3$.

Luby's Algorithm

A bad node u has outdegree of at least $2d(u)/3$.

Lemma:
At least half of all edges are good.

Recall:
An edge is *good* if it is incident on a good node.

Luby's Algorithm

A bad node u has outdegree of at least $2d(u)/3$.

Lemma:
At least half of all edges are good.

A bad edge ends in a bad node.

Recall:
An edge is *good* if it is incident on a good node.

Luby's Algorithm

A bad node u has outdegree of at least $2d(u)/3$.

Lemma:
At least half of all edges are good.

A bad edge ends in a bad node.

Recall:
An edge is *good* if it is incident on a good node.

Let B be the bad nodes.
#bad edges $\leq \Sigma_{u \in B} \text{indeg}(u)$
$\Sigma_{u \in B} \text{indeg}(u) \leq \Sigma_{u \in B} \text{outdeg}(u) / 2$

Luby’s Algorithm

A bad node u has outdegree of at least $2d(u)/3$.

Lemma:
At least half of all edges are good.

A bad edge ends in a bad node.

Recall:
An edge is *good* if it is incident on a good node.

Let B be the bad nodes.
#bad edges $\leq \Sigma_{u \in B}\, \mathrm{indeg}(u)$
$\Sigma_{u \in B}\, \mathrm{indeg}(u) \leq \Sigma_{u \in B}\, \mathrm{outdeg}(u)\ /2$

Luby's Algorithm

Lemma:
At least half of all edges are good.

Recall:
An edge is *good* if it is incident on a good node.

A bad node u has outdegree of at least $2d(u)/3$.

A bad edge ends in a bad node.

Let B be the bad nodes.
#bad edges $\leq \Sigma_{u \in B}\, \mathrm{indeg}(u)$
$\Sigma_{u \in B}\, \mathrm{indeg}(u) \leq \Sigma_{u \in B}\, \mathrm{outdeg}(u)\ /2$

#bad edges $\leq \Sigma_{u \in B}\, \mathrm{outdeg}(u)\ /2$
$\leq$ #edges/2

Luby’s Algorithm

Lemma:
A good edge gets removed with a constant probability.

Lemma:
At least half of all edges are good.

Corollary:
After expected $O(\log m)$ rounds, no edges remain.
We can get this in expectation and w.h.p.

Luby's Algorithm

Lemma:
A good edge gets removed with probability at least 1/36.

Lemma:
At least half of all edges are good.

Linearity of expectation:
Let R be the number of edges removed in a phase.

$$\boldsymbol{E}[R] \geq m/72$$

Markov's inequality:

$$P\left[R \leq \frac{\boldsymbol{E}[R]}{2}\right] \leq \frac{1}{144}$$

Luby's Algorithm

Lemma:
A good edge gets removed with probability at least 1/36.

Lemma:
At least half of all edges are good.

Let R be the number of edges removed in a phase.

$$P\left[R \leq \frac{\boldsymbol{E}[R]}{2}\right] \leq \frac{1}{144}$$

Let $\hat{m}$ be the number of edges in the original input graph.

After expected $O(\log \hat{m}) = O(\log n)$ rounds, all edges are removed. Notice that a degree 0 node joins the MIS with probability 1.

Luby's Algorithm

Lemma:
A good edge gets removed with a constant probability.

Lemma:
At least half of all edges are good.

Corollary:
After expected $O(\log m)$ rounds, no edges remain. We can get this in expectation and w.h.p.

Left as an exercise.

Literature

Maximal Independent Set

Deterministic:

$O(\text{poly} \log n)$

Randomized:

[Ghaffari et al., SODA 2021]

Literature

Maximal Independent Set

Deterministic:

$O(\text{poly} \log n)$

$\Omega(\log n)$

Randomized:

$\Omega(\log \log n)$

[Ghaffari et al., SODA 2021]

[Balliu et al., FOCS 2019 best paper]

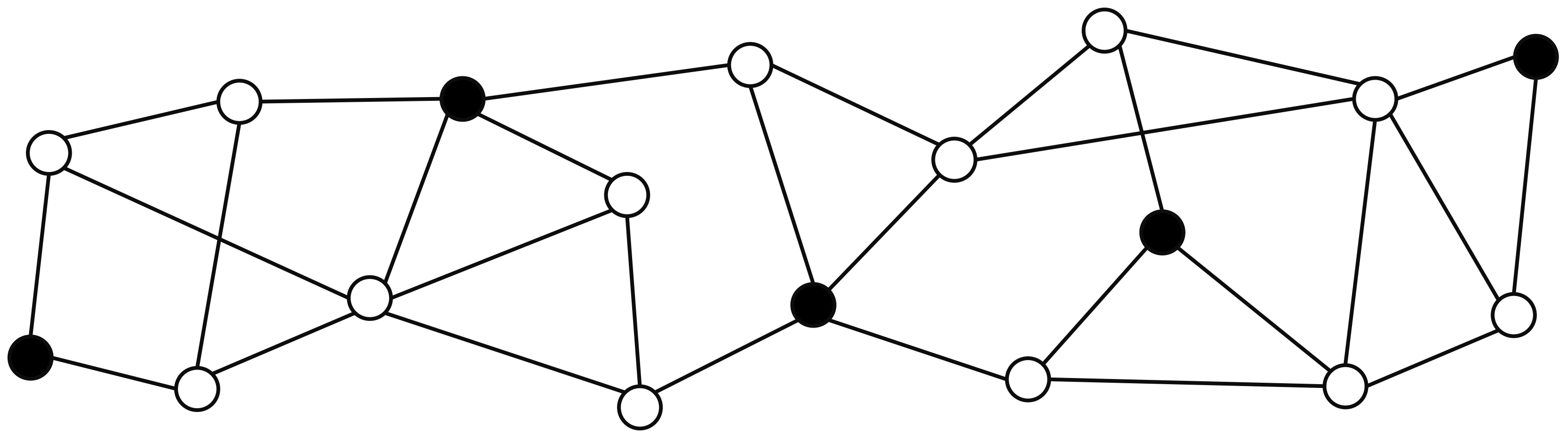
Literature

Maximal Independent Set
Deterministic:
$O(\text{poly}\log n)$
$\Omega(\log n)$

Randomized:
$\mathrm{O}(\log \Delta + \text{poly}\log\log n)$
$\Omega(\log\log n)$

[Ghaffari et al., SODA 2021]
[Balliu et al., FOCS 2019 best paper]
[Ghaffari, SODA 2016 best student paper]

Wrap-up



Luby's Algorithm: $O(\log n)$

Maximal Independent Set

Deterministic:

$O(\text{poly} \log n)$

$\Omega(\log n)$

Randomized:

$O\left(\sqrt{\log \Delta} + \text{poly} \log \log n\right)$

$\Omega(\log \log n)$

Maximum Independent Set

☹