Graph Property Testing and the Container Method

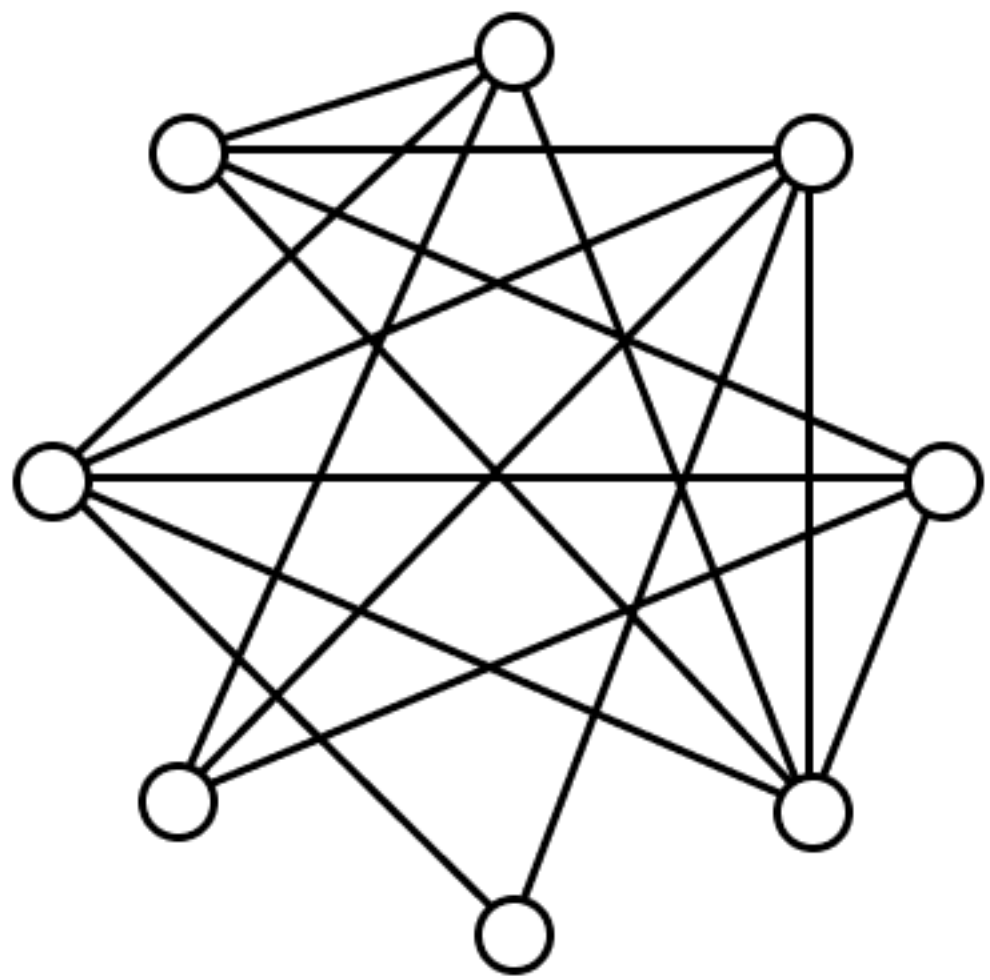
Cameron Seth

September 8 2026

UNIVERSITY OF WATERLOO

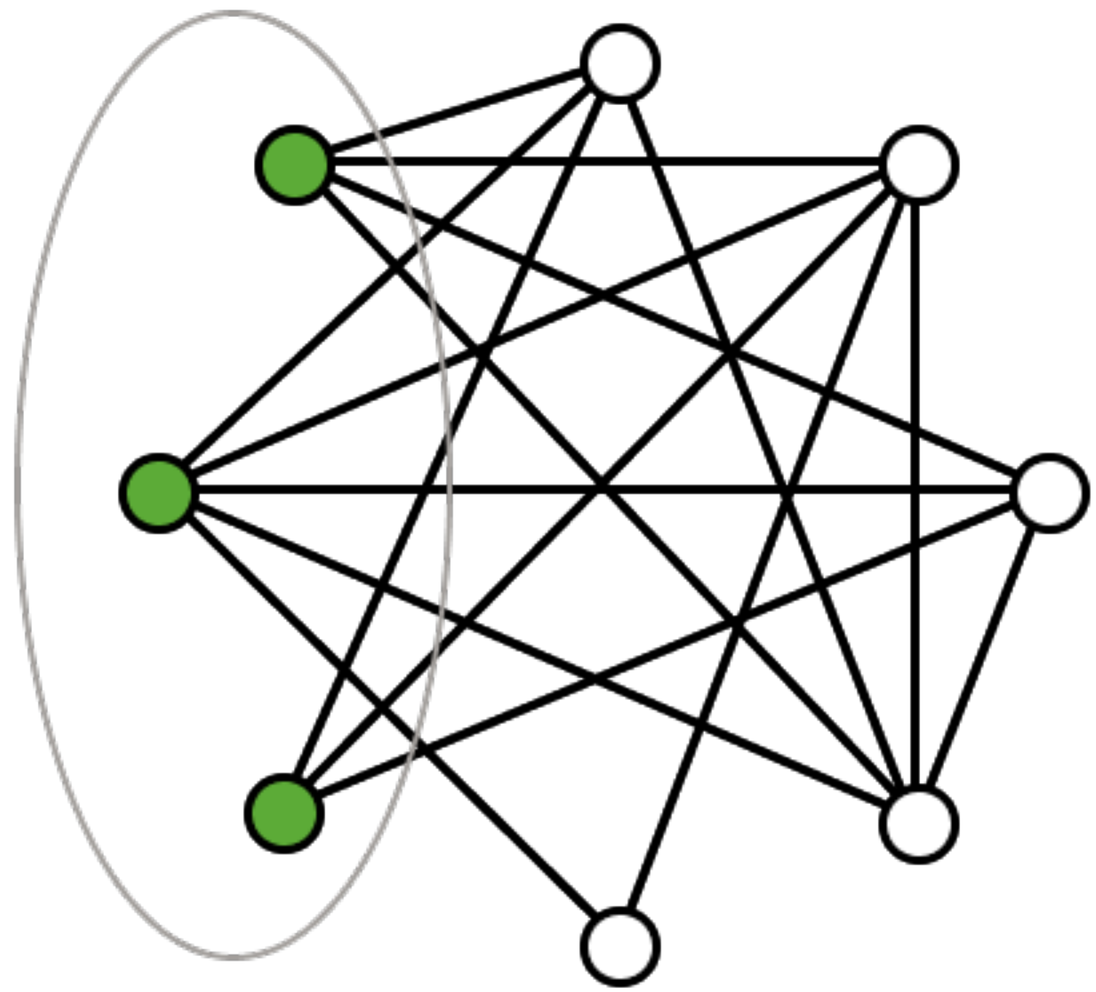
Independent Set Problem

Given a graph G on n vertices, does it have an independent set of size ρn?



Independent Set Problem

Given a graph G on n vertices, does it have an independent set of size ρn?

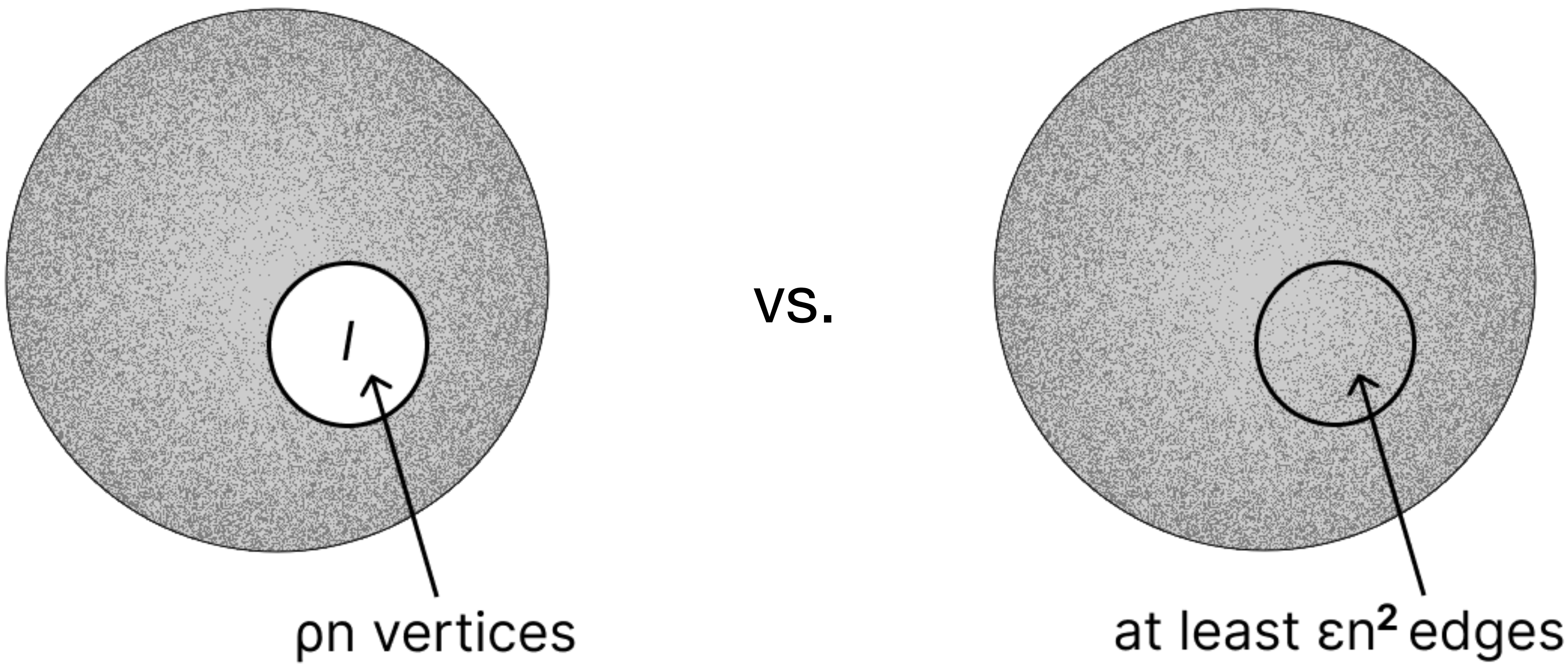


Testing Independent Sets

Problem: Distinguish between the cases:

(i) G has a ρn independent set, and

(ii) every induced subgraph of size ρn has at least ϵn^2 edges



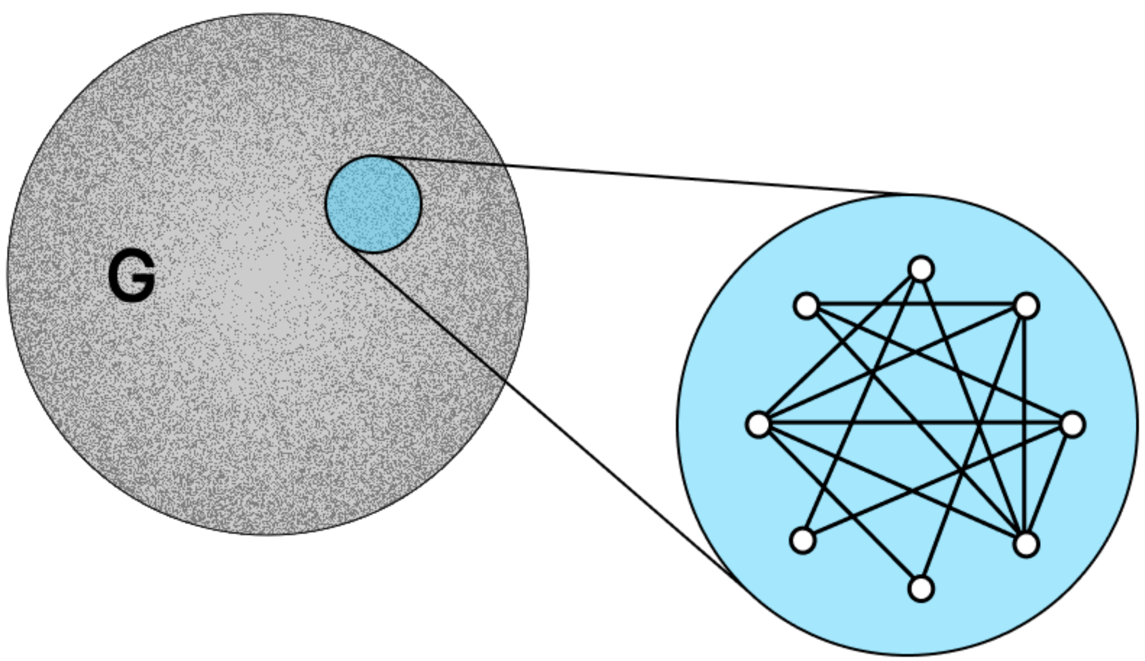
Theorem: Inspecting a random subgraph on $\tilde{O}(\rho/\epsilon^4)$ vertices is sufficient for distinguishing between (i) and (ii) (whp).

[Goldreich, Goldwasser, Ron '98]

Definitions

An ϵ-tester for Π is an algorithm that samples a set S of s random vertices, examines the induced subgraph $G[S]$, and distinguishes between the cases (with probability at least $2/3$):

1. G has the property Π, and
2. At least ϵn^2 edges of G must be modified to make G have Π (ϵ-far)



s is the sample complexity of the tester.

First Results in Graph Property Testing

Theorem: There are ϵ-testers for ρ-independent set, k-colorability, and many other partition properties with sample complexity $\mathrm{poly}(1/\epsilon)$.

[Goldreich, Goldwasser, Ron '98]

First Results in Graph Property Testing

Theorem: There are ϵ-testers for ρ-independent set, k-colorability, and many other partition properties with sample complexity $\mathrm{poly}(1/\epsilon)$.

[Goldreich, Goldwasser, Ron '98]

Question: What is the minimum sample complexity of testing these graph properties?

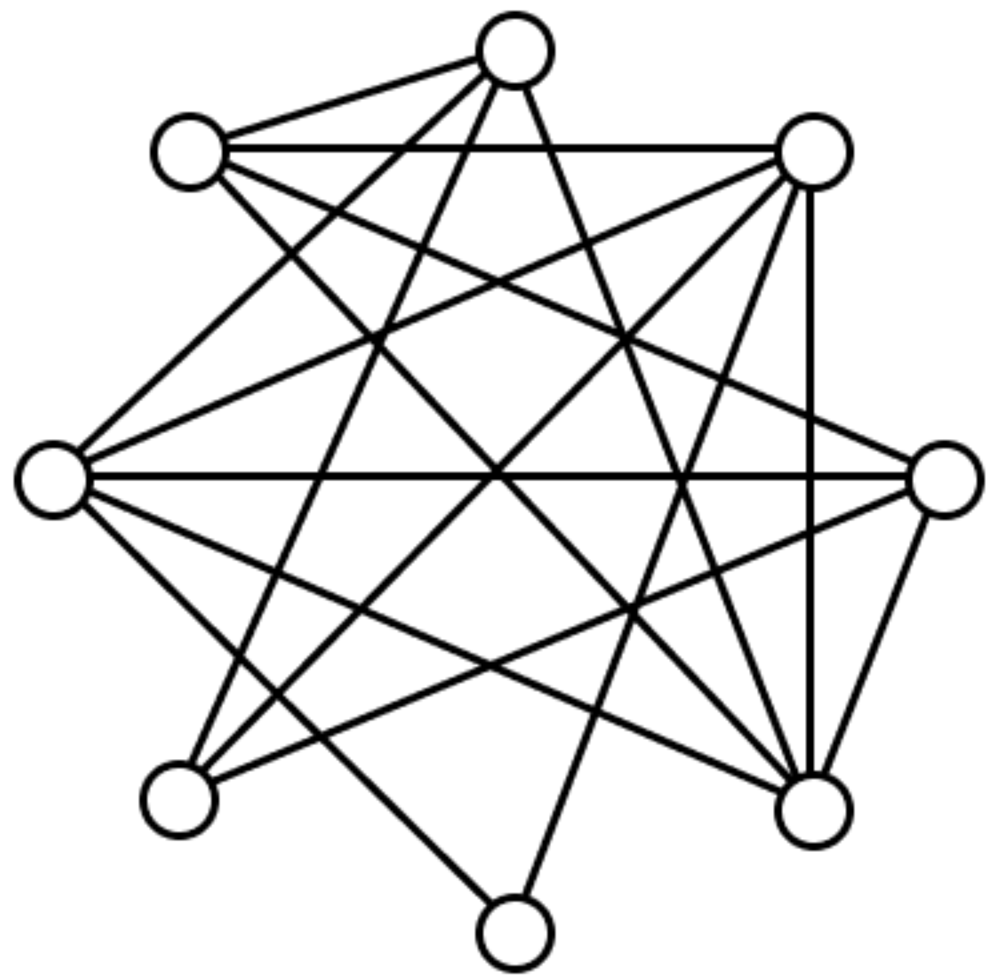
This Thesis: The container method provides a general framework for proving strong upper bounds on the sample complexity of testing various graph and hypergraph properties.

1. Tight bounds on testing the ρ-independent set property
2. New upper bounds on testing satisfiability of CSPs (implies new upper bounds for $k-$colorability and other partition properties)
3. Tight bounds on tolerant testing the ρ-independent set property

What is the Container Method?

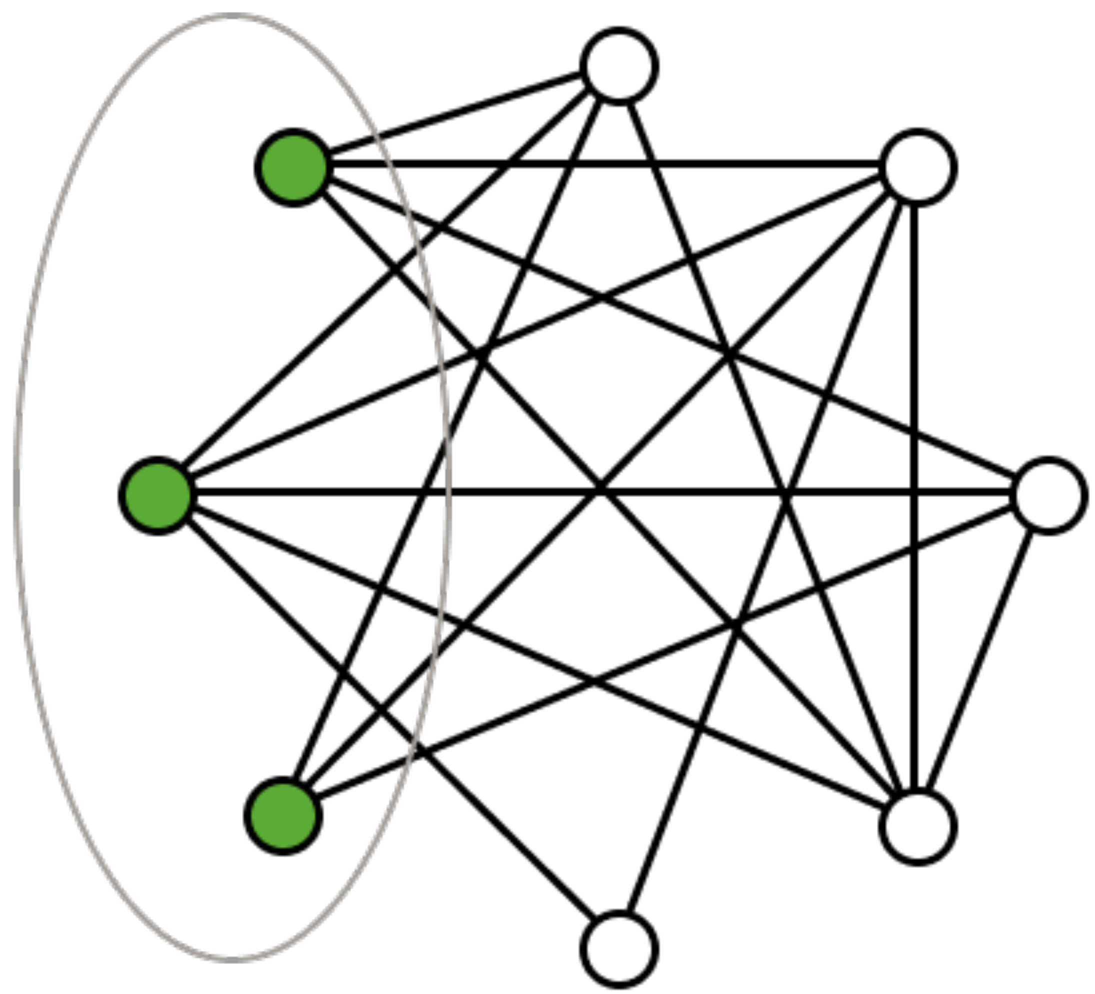
What is the Container Method?

Answer: A tool for characterizing independent sets in some graphs.



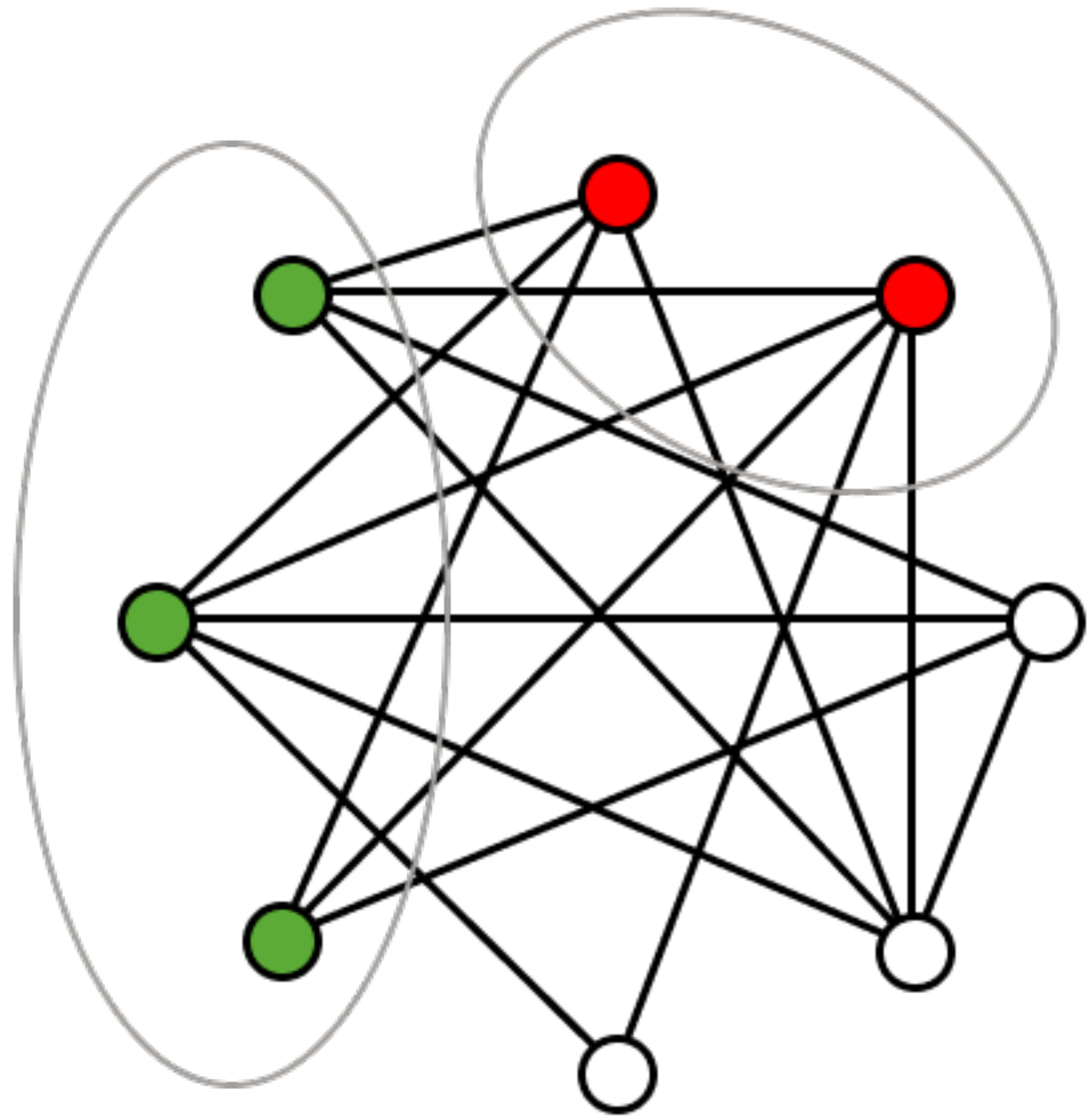
What is the Container Method?

Answer: A tool for characterizing independent sets in some graphs.



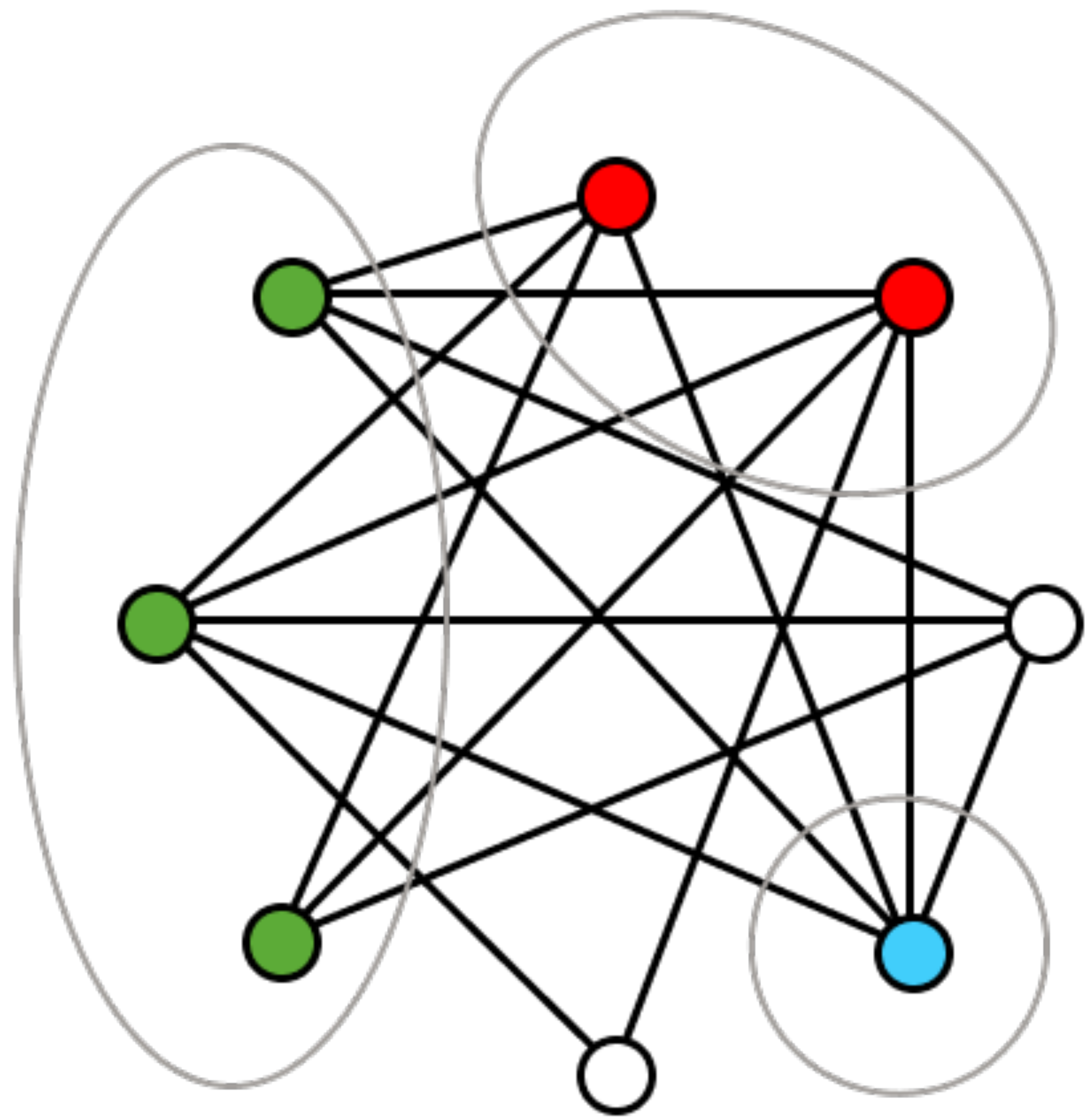
What is the Container Method?

Answer: A tool for characterizing independent sets in some graphs.



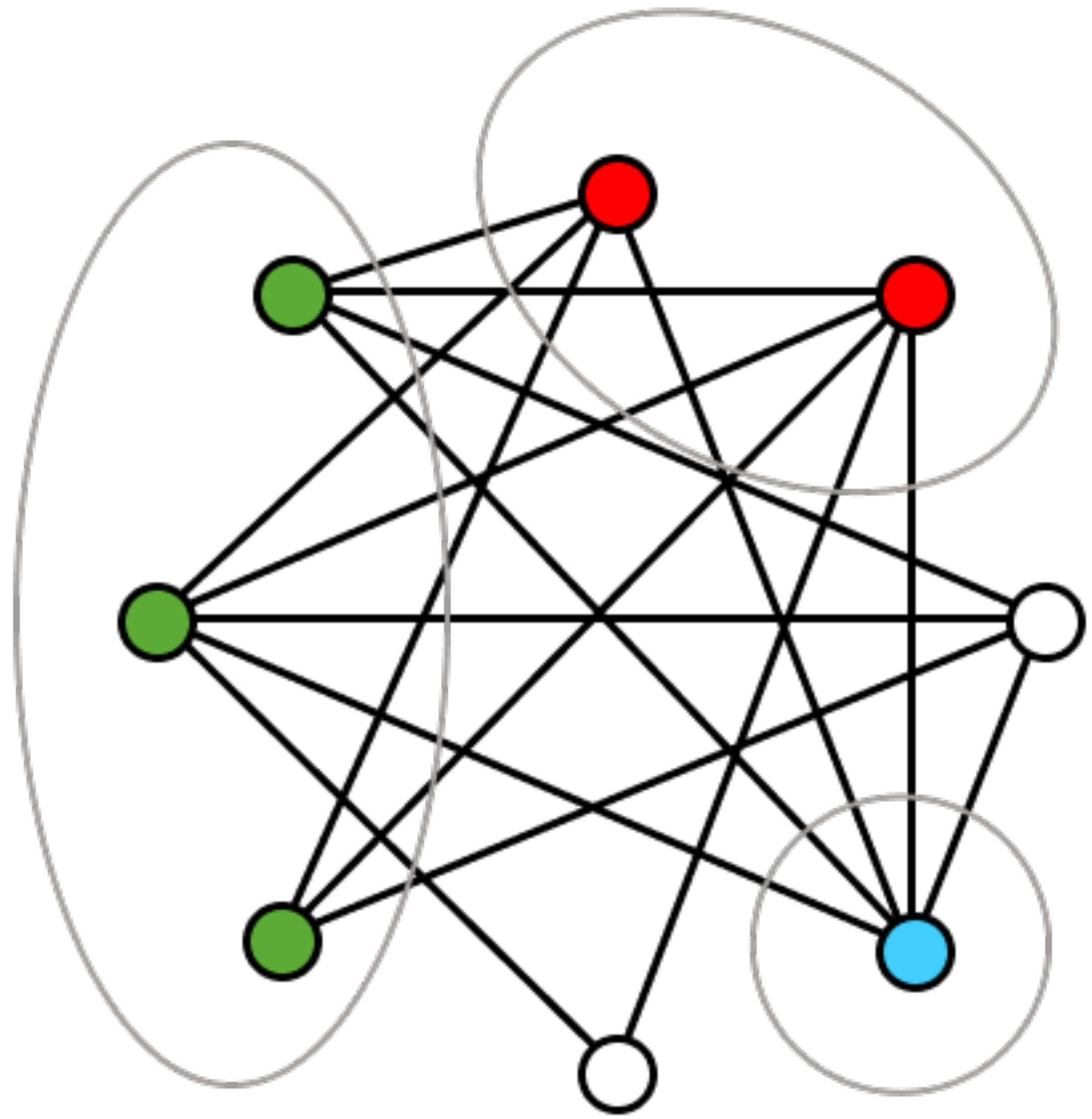
What is the Container Method?

Answer: A tool for characterizing independent sets in some graphs.



What is the Container Method?

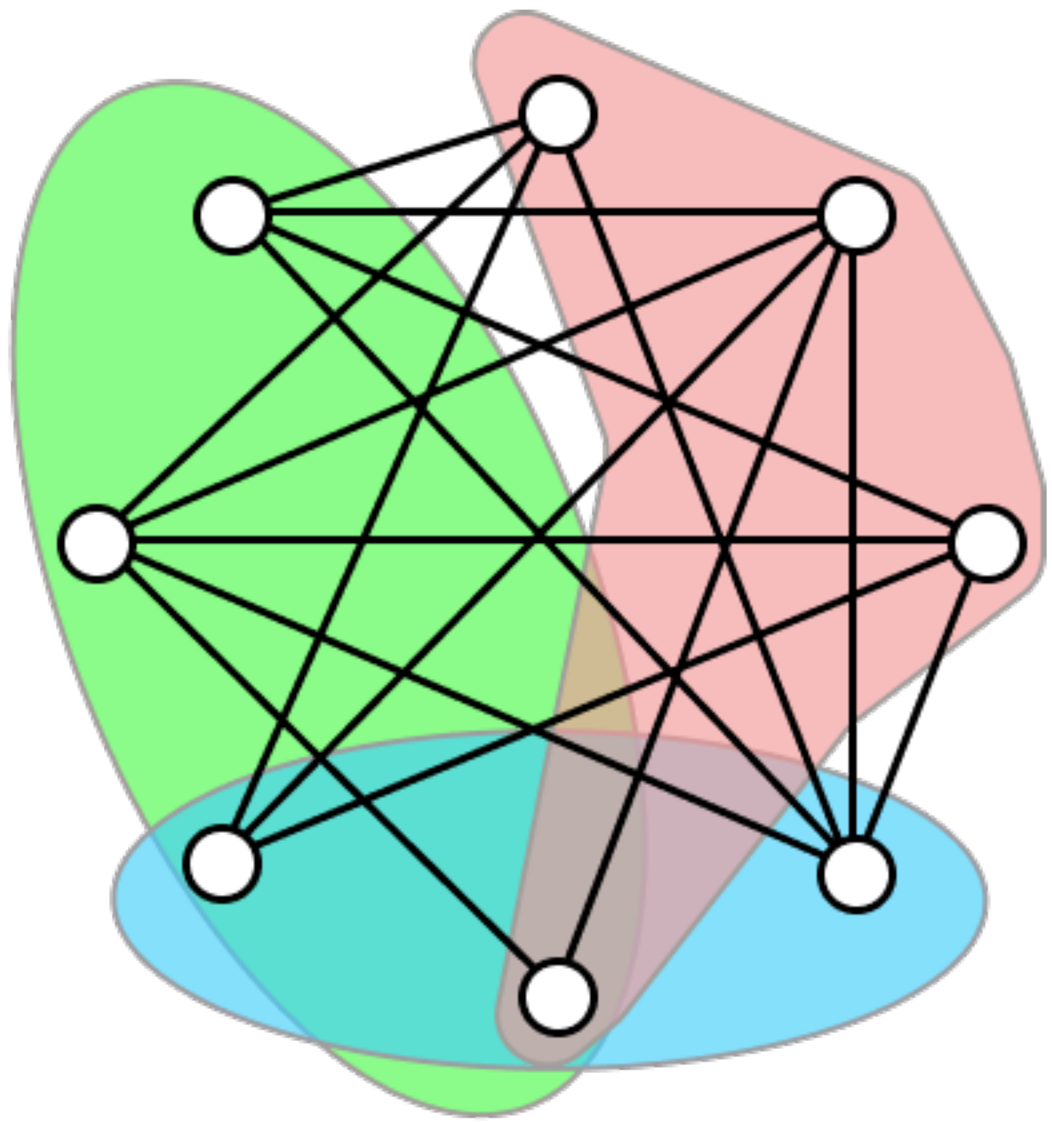
Answer: A tool for characterizing independent sets in some graphs.



Informal Idea: For any graph satisfying some "nice" conditions, all independent sets in the graph can be covered by a small number of containers (each container is a subset of vertices).

What is the Container Method?

Answer: A tool for characterizing independent sets in some graphs.



Informal Idea: For any graph satisfying some "nice" conditions, all independent sets in the graph can be covered by a small number of containers (each container is a subset of vertices).

An Initial Graph Container Lemma

Lemma: For any ϵ, ρ let $G = (V, E)$ be a graph such that every induced subgraph on ρn vertices has at least ϵn^2 edges. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

1. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
2. for every $C \in \mathscr{C}$, $|C| \leq (1 - \epsilon/2)\rho n$.
3. for every independent set I, there exists $C \in \mathscr{C}$ with $I \subseteq C$.

[Kleitman, Winston '82]

An Initial Graph Container Lemma

Lemma: For any ϵ, ρ let $G = (V, E)$ be a graph such that every induced subgraph on ρn vertices has at least ϵn^2 edges. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

1. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
2. for every $C \in \mathscr{C}$, $|C| \leq (1 - \epsilon/2)\rho n$.
3. for every independent set I, there exists $C \in \mathscr{C}$ with $I \subseteq C$.

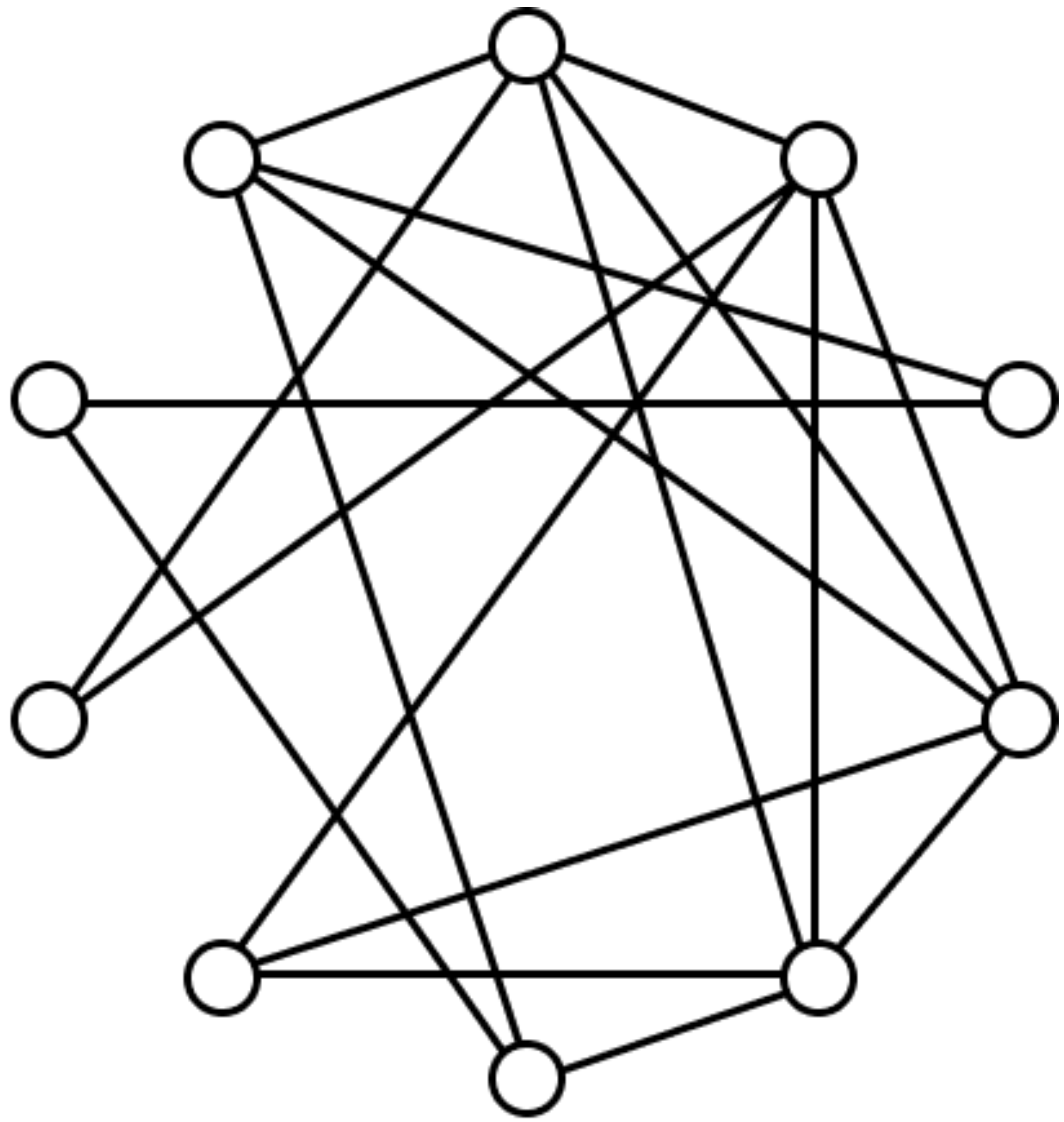
[Kleitman, Winston '82]

Note: for survey of combinatorial applications see "Counting Independent Sets in Graphs" by Samotij or "The method of hypergraph containers" by Balogh, Morris, and Samotij.

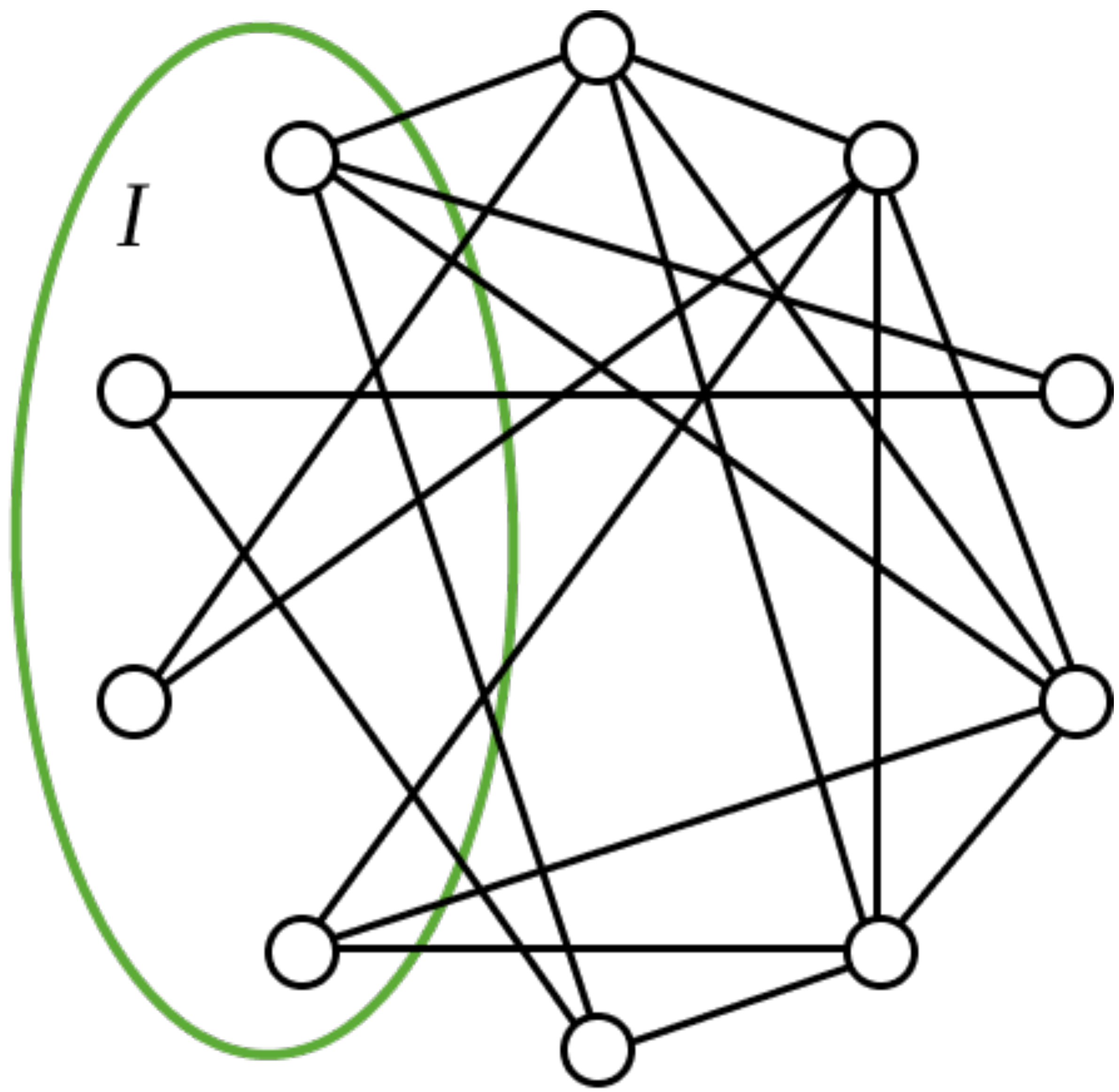
How to prove a Container Lemma -
An Encoding Argument

Encoding Independent Sets

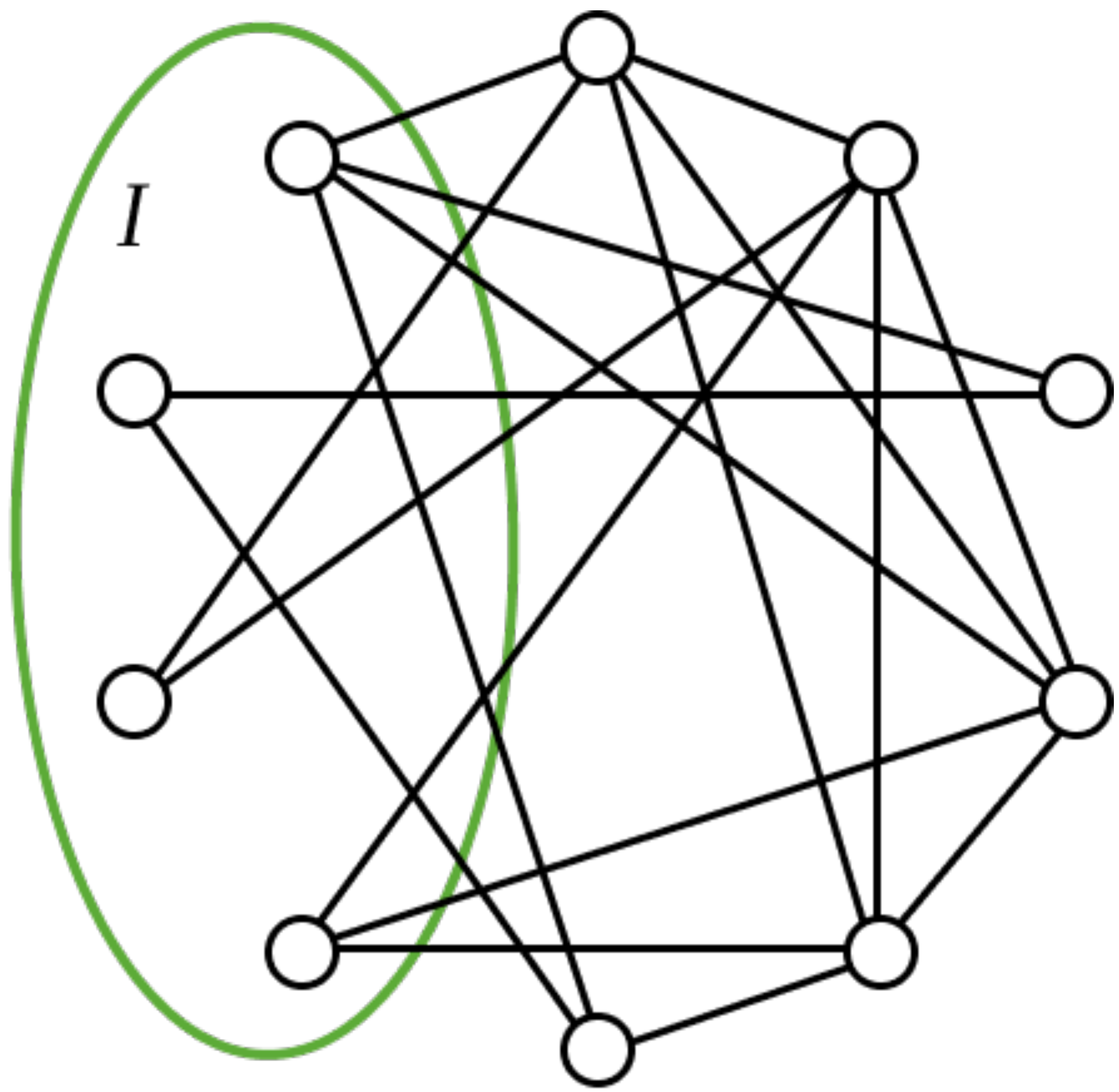
Encoding Independent Sets



Encoding Independent Sets

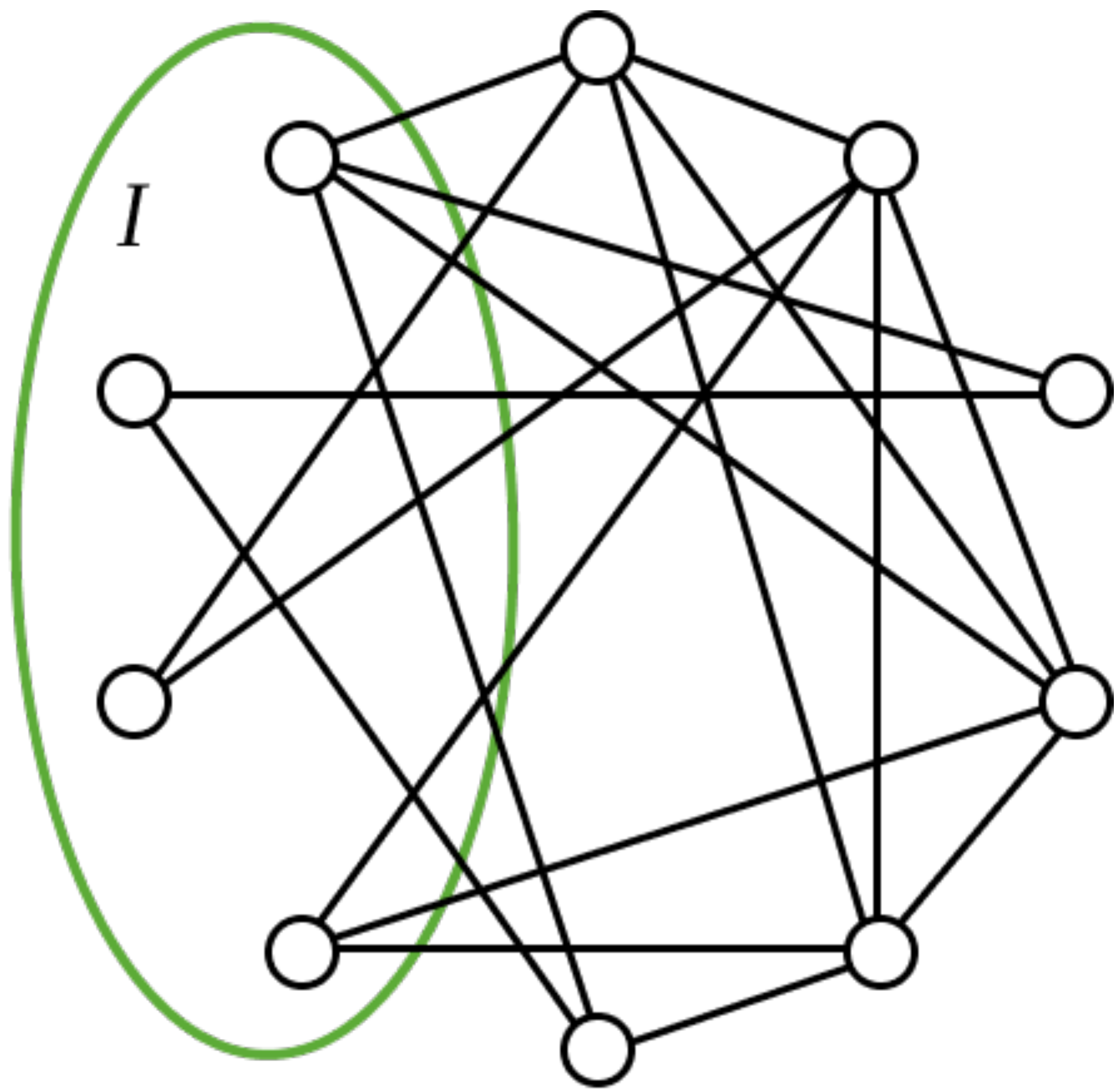


Encoding Independent Sets



How can I give you the most information about an independent set I by just telling you about one of the vertices in I?

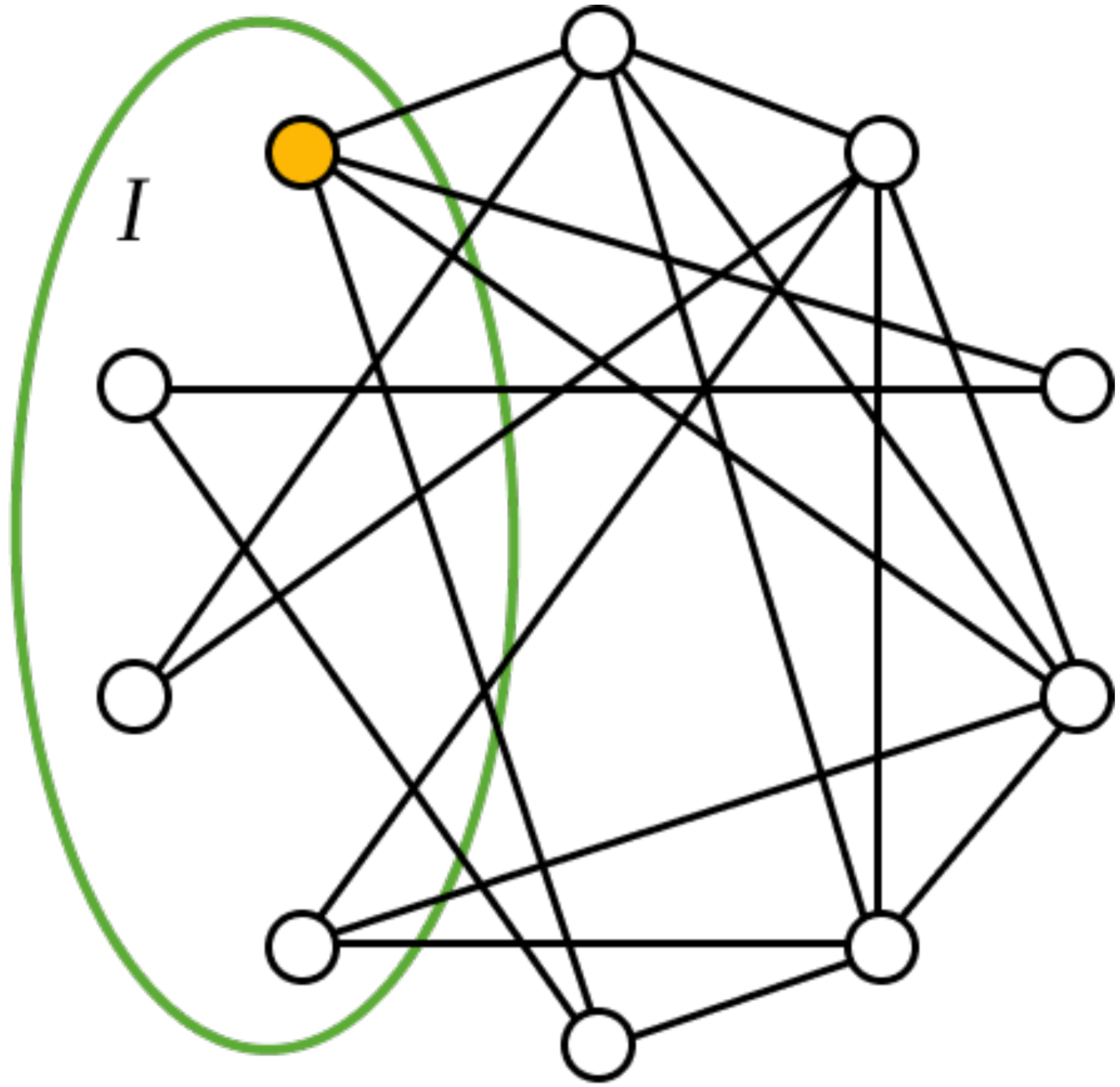
Encoding Independent Sets



How can I give you the most information about an independent set I by just telling you about one of the vertices in I?

Answer: Send the vertex $v \in I$ with highest degree.

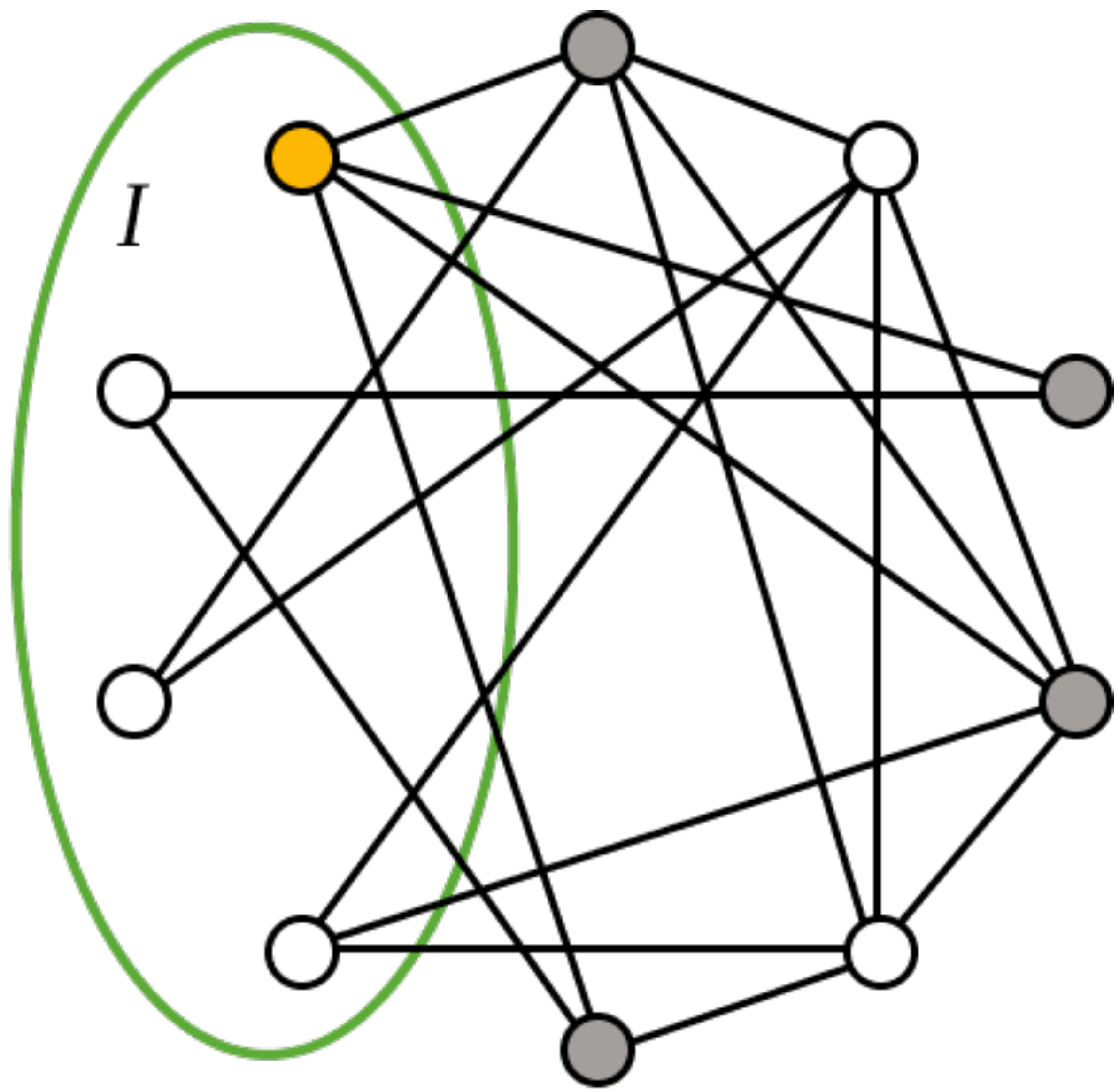
Encoding Independent Sets



How can I give you the most information about an independent set I by just telling you about one of the vertices in I?

Answer: Send the vertex $v \in I$ with highest degree.

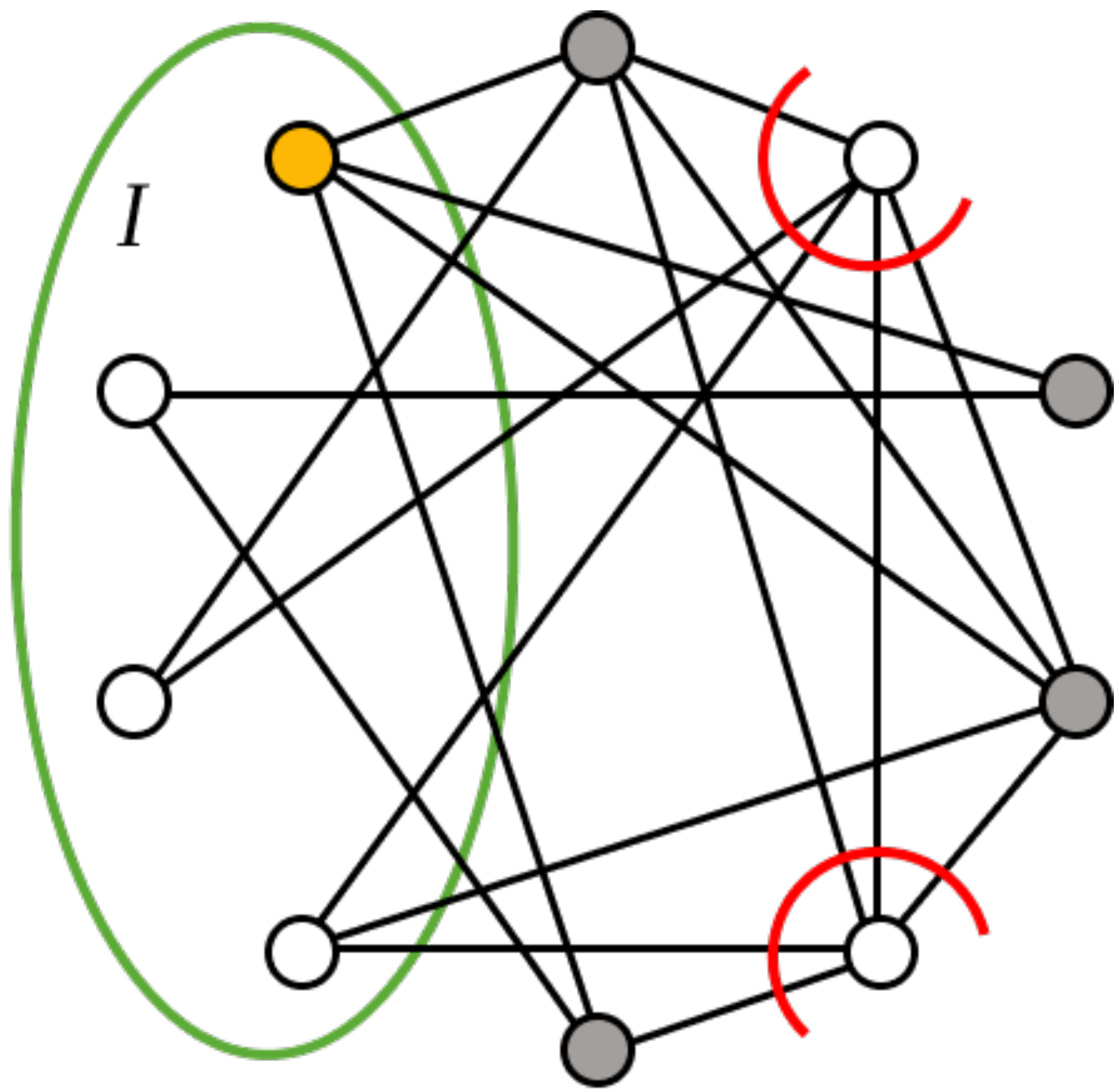
Encoding Independent Sets



How can I give you the most information about an independent set I by just telling you about one of the vertices in I?

Answer: Send the vertex $v \in I$ with highest degree.

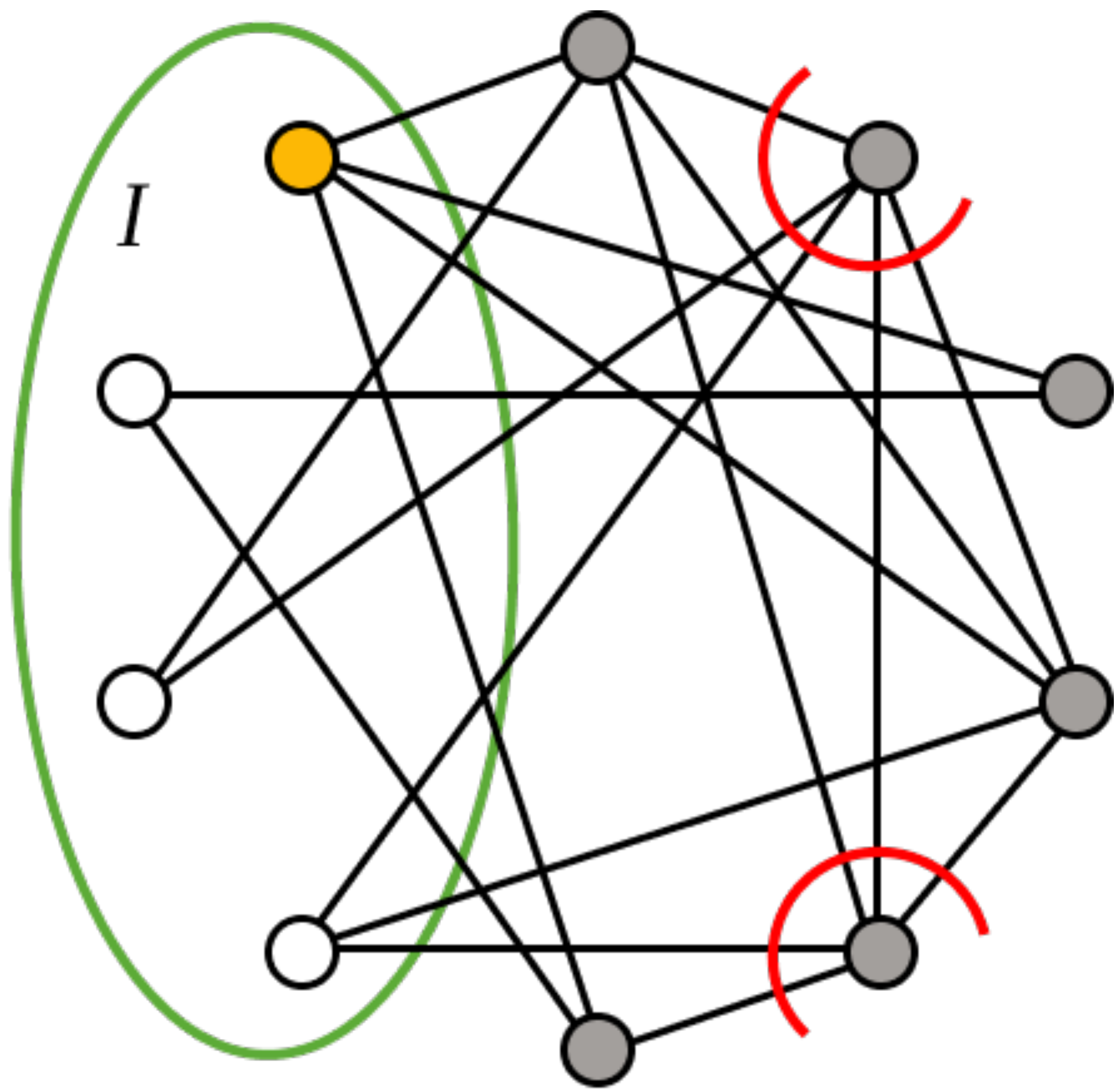
Encoding Independent Sets



How can I give you the most information about an independent set I by just telling you about one of the vertices in I?

Answer: Send the vertex $v \in I$ with highest degree.

Encoding Independent Sets



How can I give you the most information about an independent set I by just telling you about one of the vertices in I?

Answer: Send the vertex $v \in I$ with highest degree.

Container Generator Algorithm

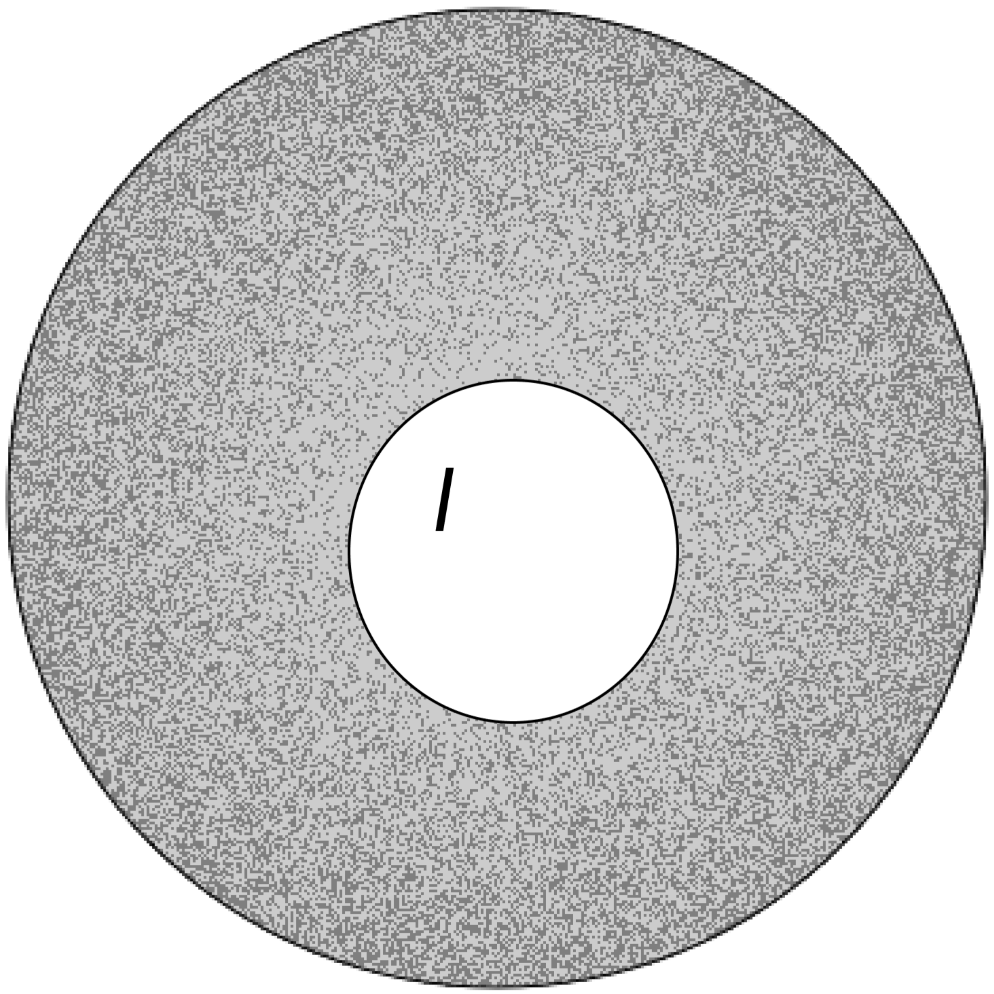
Input: Graph G and an independent set I

Output: Container $C(I)$ and fingerprint $F(I)$

Container Generator Algorithm

Input: Graph G and an independent set I

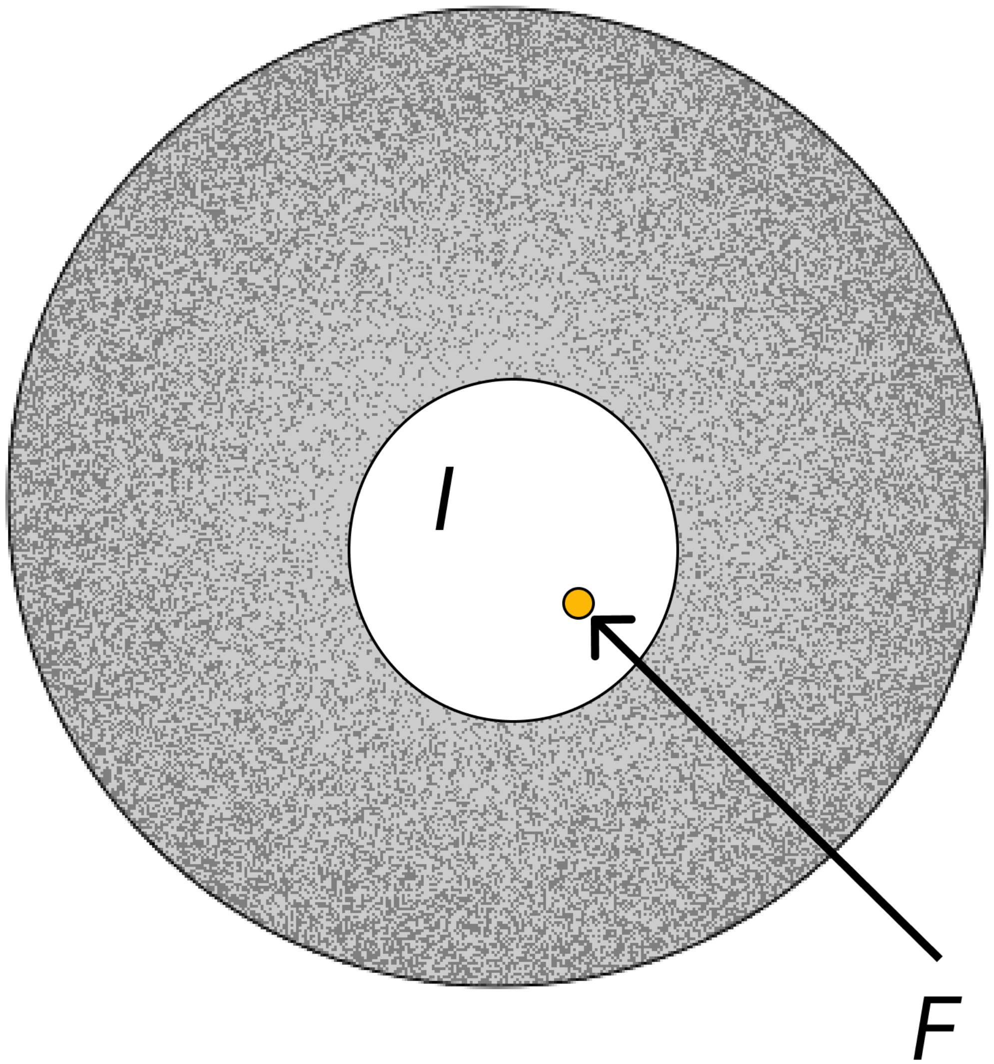
Output: Container $C(I)$ and fingerprint $F(I)$



Container Generator Algorithm

Input: Graph G and an independent set I

Output: Container $C(I)$ and fingerprint $F(I)$

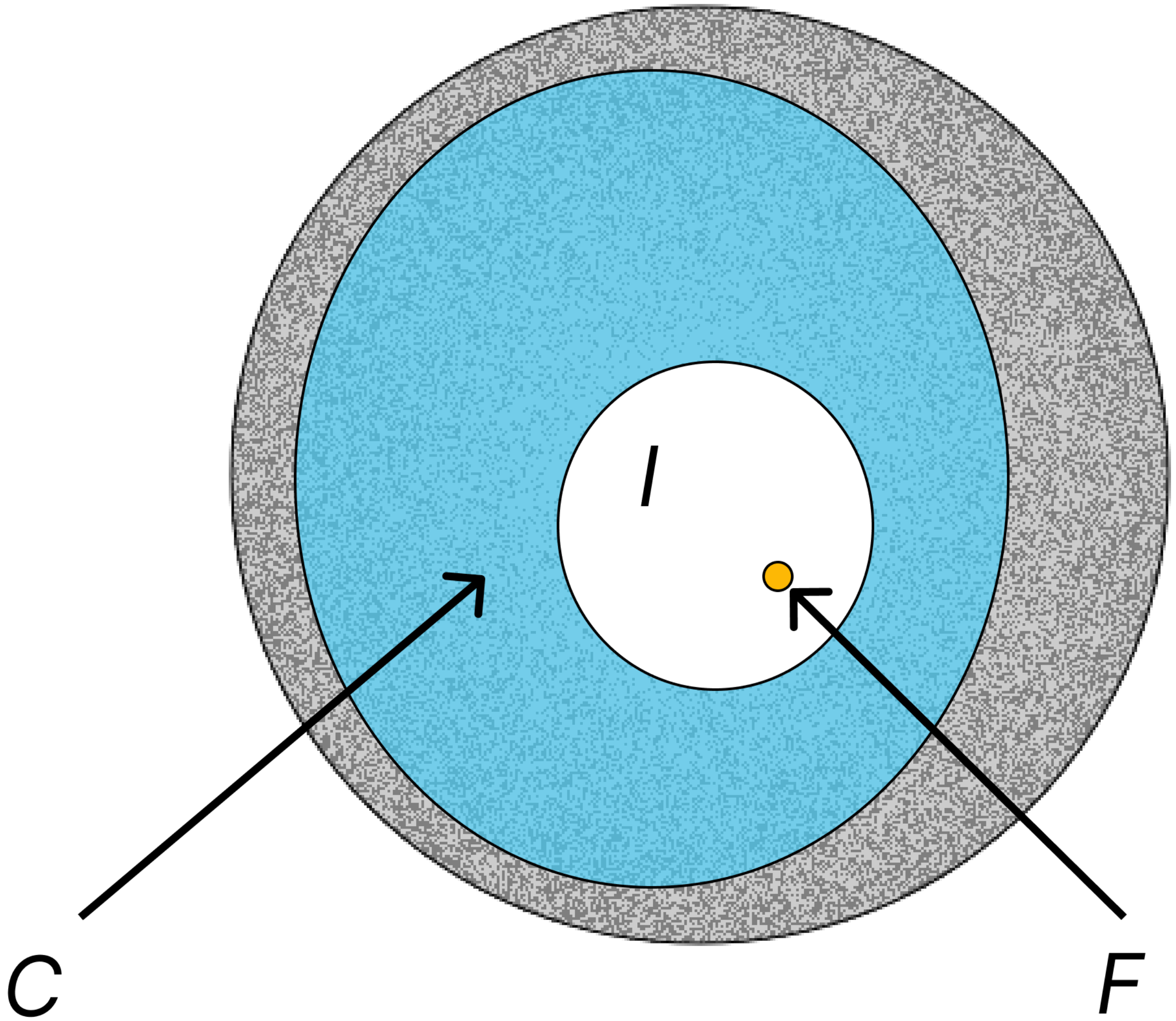


1st Iteration

Container Generator Algorithm

Input: Graph G and an independent set I

Output: Container $C(I)$ and fingerprint $F(I)$

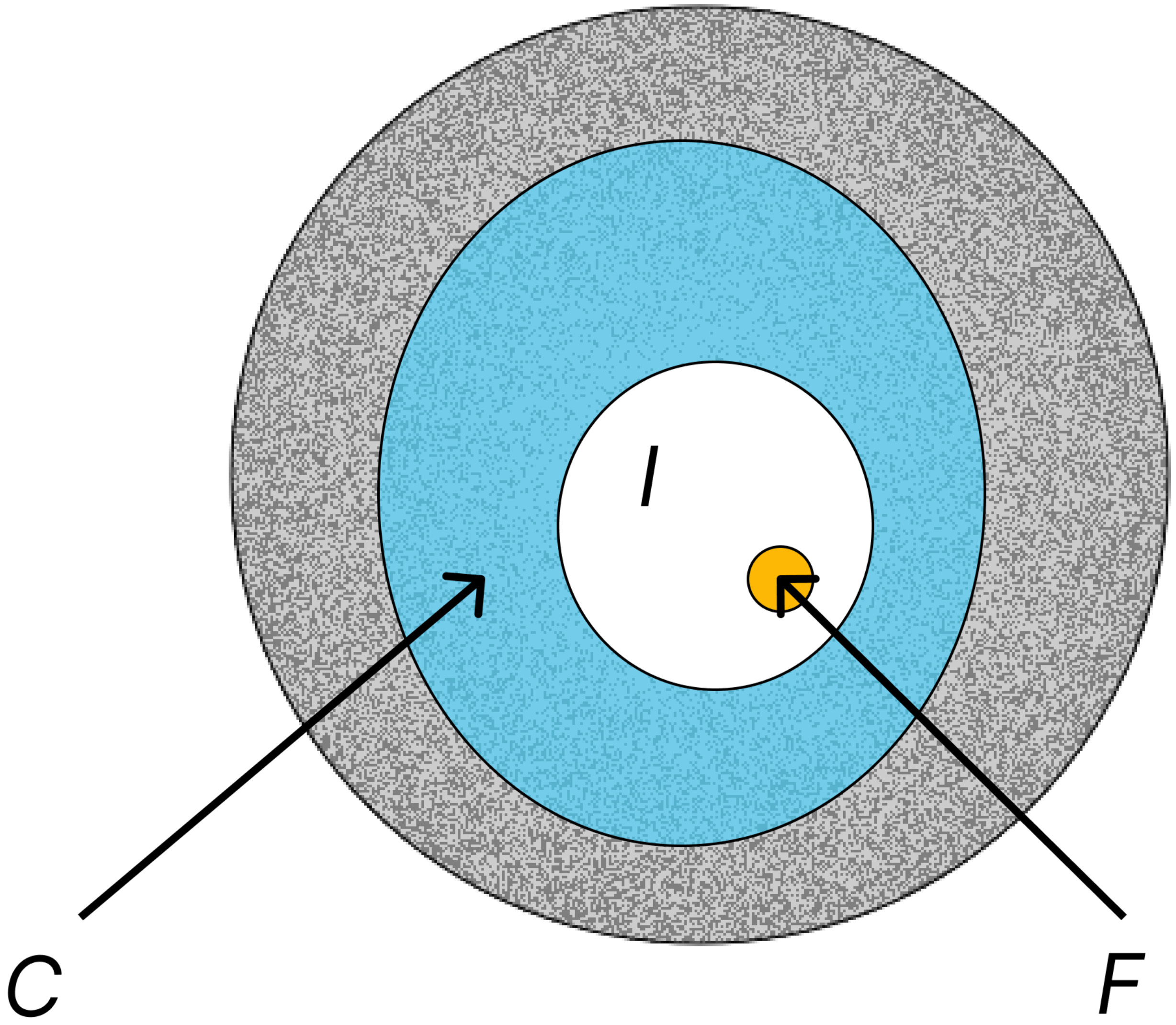


1st Iteration

Container Generator Algorithm

Input: Graph G and an independent set I

Output: Container $C(I)$ and fingerprint $F(I)$

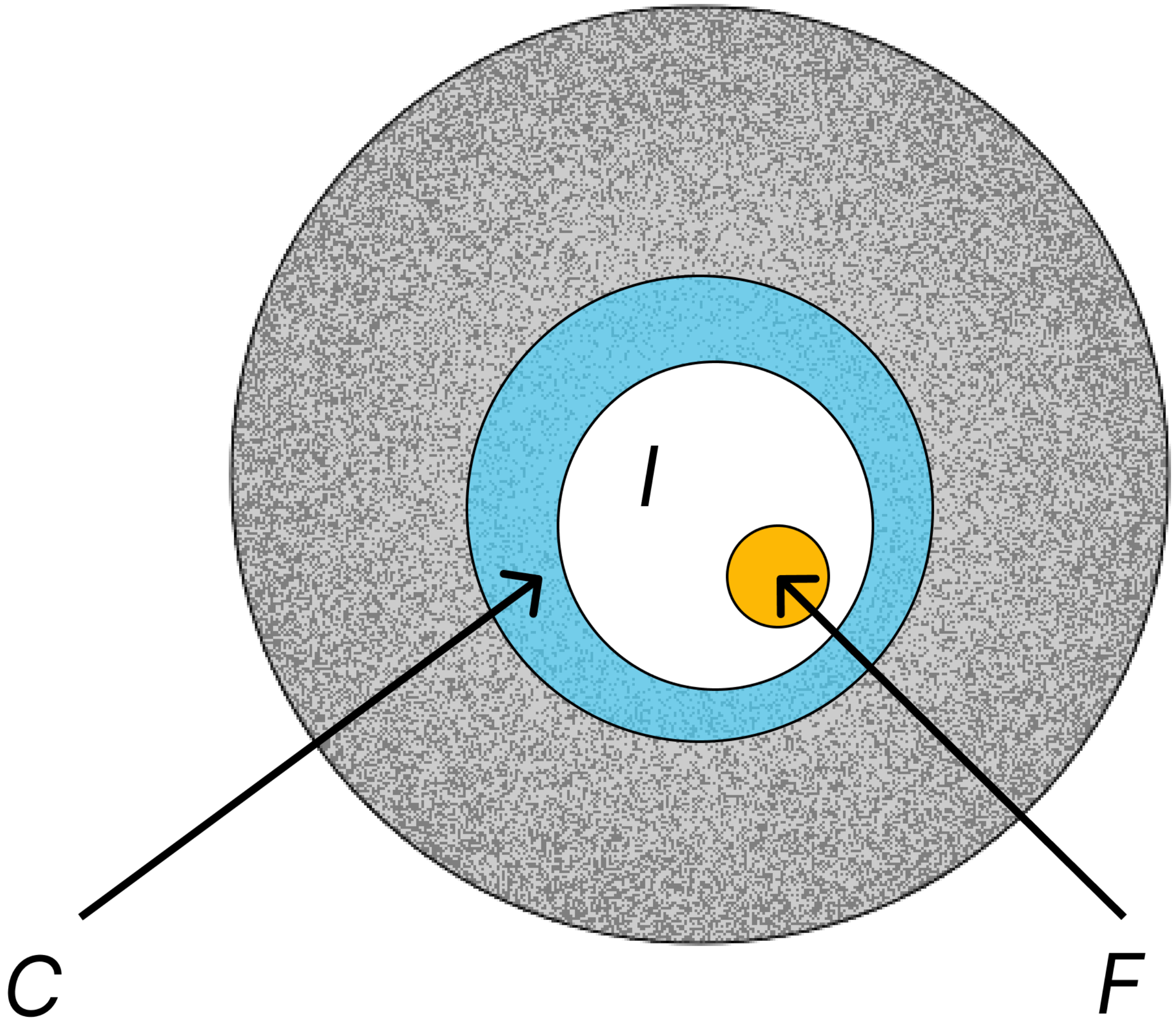


2nd Iteration

Container Generator Algorithm

Input: Graph G and an independent set I

Output: Container $C(I)$ and fingerprint $F(I)$

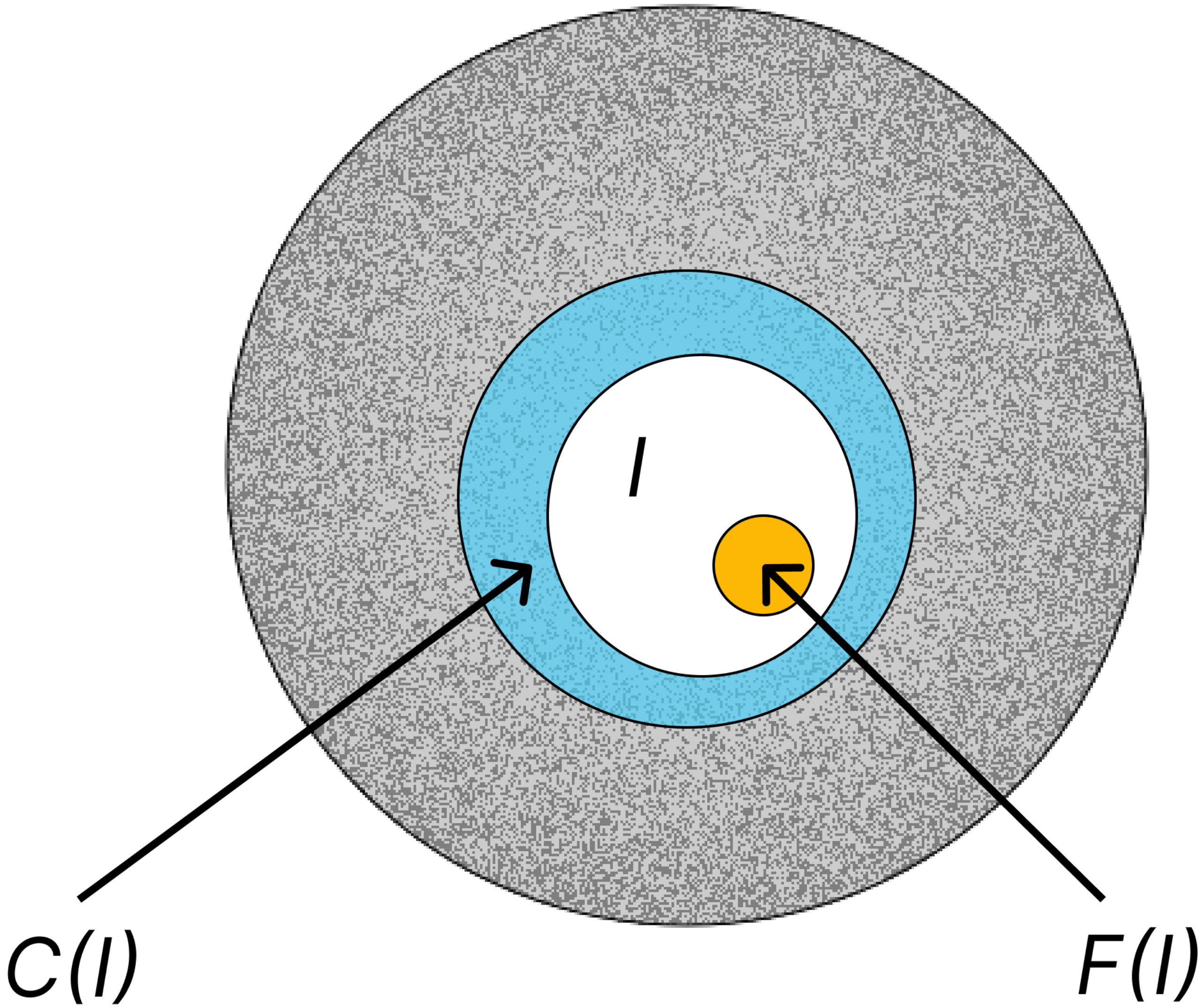


Final Iteration

Container Generator Algorithm

Input: Graph G and an independent set I

Output: Container $C(I)$ and fingerprint $F(I)$

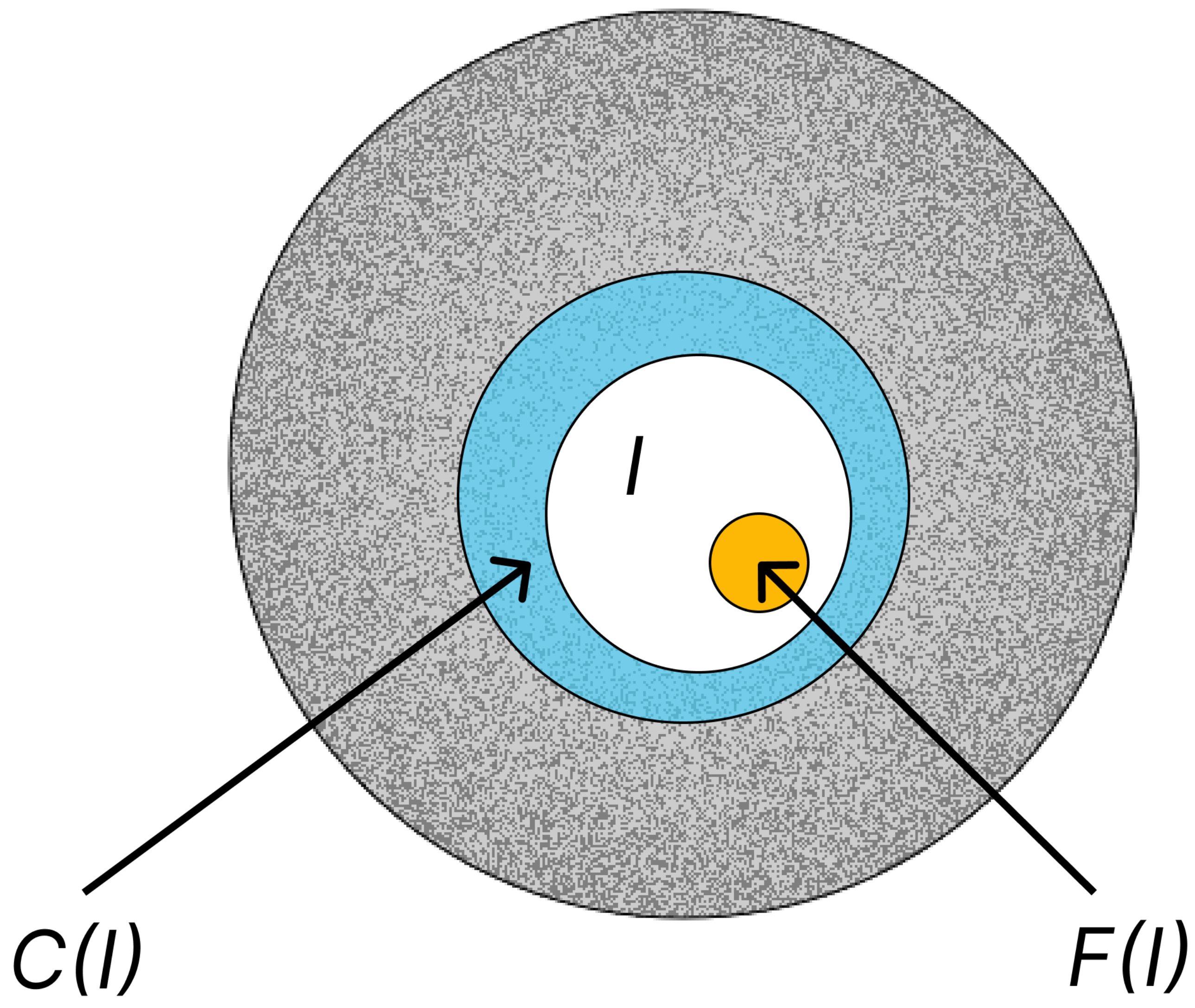


Final Iteration

Container Generator Algorithm

Input: Graph G and an independent set I

Output: Container $C(I)$ and fingerprint $F(I)$



Final Iteration

Key Observations:

- $I \subseteq C(I)$
- $C(I)$ can be reconstructed from $F(I)$

Proving the Simple Graph Container Lemma

Lemma: For any ϵ, ρ let $G = (V, E)$ be a graph such that every induced subgraph on ρn vertices has at least ϵn^2 edges. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

1. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
2. for every $C \in \mathscr{C}$, $|C| \leq (1 - \epsilon/2)\rho n$.
3. for every independent set I, there exists $C \in \mathscr{C}$ with $I \subseteq C$.

[Kleitman, Winston '82]

Proving the Simple Graph Container Lemma

Lemma: For any ϵ, ρ let $G = (V, E)$ be a graph such that every induced subgraph on ρn vertices has at least ϵn^2 edges. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

1. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
2. for every $C \in \mathscr{C}$, $|C| \leq (1 - \epsilon/2)\rho n$.
3. for every independent set I, there exists $C \in \mathscr{C}$ with $I \subseteq C$.

[Kleitman, Winston ’82]

Proof Idea: How do we construct $\mathscr{C}$?

Proving the Simple Graph Container Lemma

Lemma: For any ϵ, ρ let $G = (V, E)$ be a graph such that every induced subgraph on ρn vertices has at least ϵn^2 edges. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

1. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
2. for every $C \in \mathscr{C}$, $|C| \leq (1 - \epsilon/2)\rho n$.
3. for every independent set I, there exists $C \in \mathscr{C}$ with $I \subseteq C$.

[Kleitman, Winston '82]

Proof Idea: How do we construct $\mathscr{C}$?

Run the container procedure on every independent set for $1/\epsilon$ steps

Let $\mathscr{C} = \{C(I) : I$ is independent set in $G\}$

Proving the Simple Graph Container Lemma

Lemma: For any ϵ, ρ let $G = (V, E)$ be a graph such that every induced subgraph on ρn vertices has at least ϵn^2 edges. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

1. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
2. for every $C \in \mathscr{C}$, $|C| \leq (1 - \epsilon/2)\rho n$.
3. for every independent set I, there exists $C \in \mathscr{C}$ with $I \subseteq C$.

[Kleitman, Winston '82]

Proof Idea: How do we construct $\mathscr{C}$?

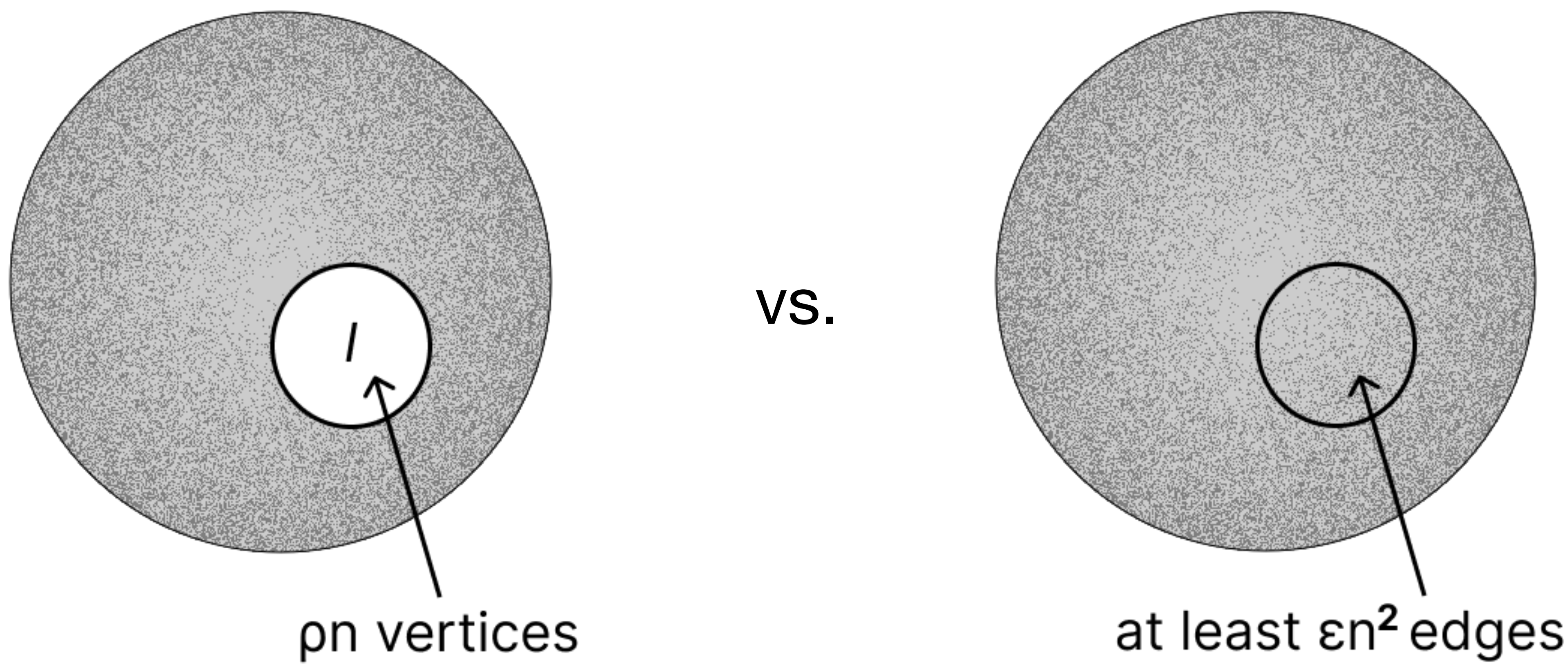
Run the container procedure on every independent set for $1/\epsilon$ steps

Let $\mathscr{C} = \{C(I) : I \text{ is independent set in } G\}$

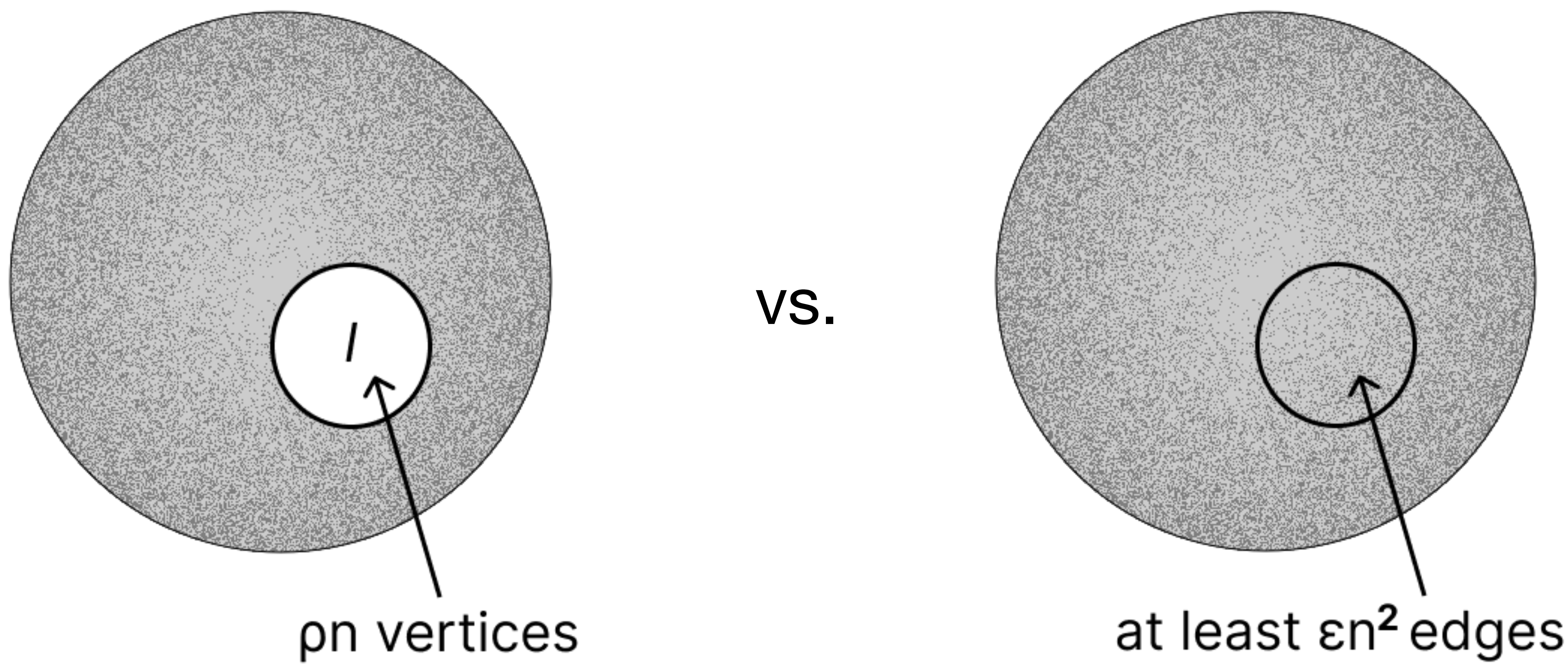
Each $C(I)$ can be reconstructed from $F(I)$, and there aren't too many possible fingerprints!

How to use Container Lemmas for Property Testing

Application to Testing ρ-Independent Set

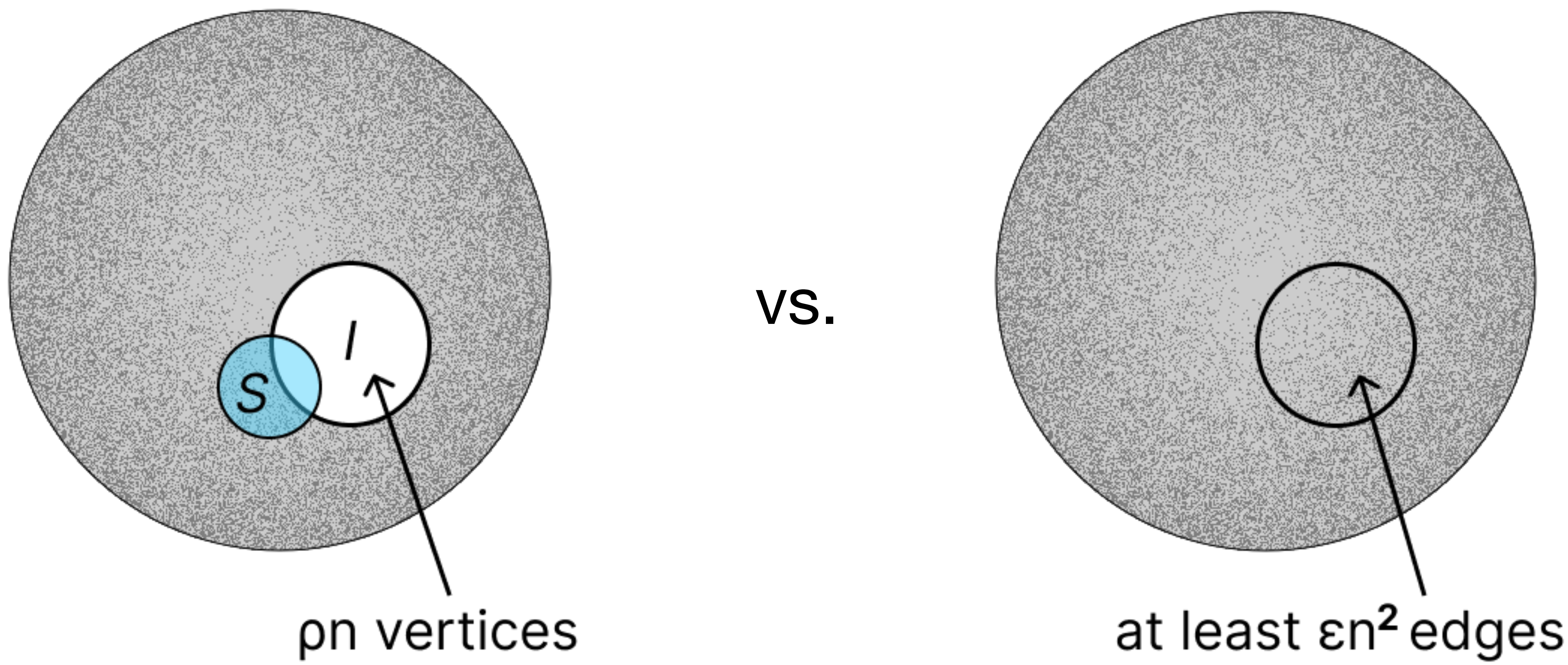


Application to Testing ρ-Independent Set



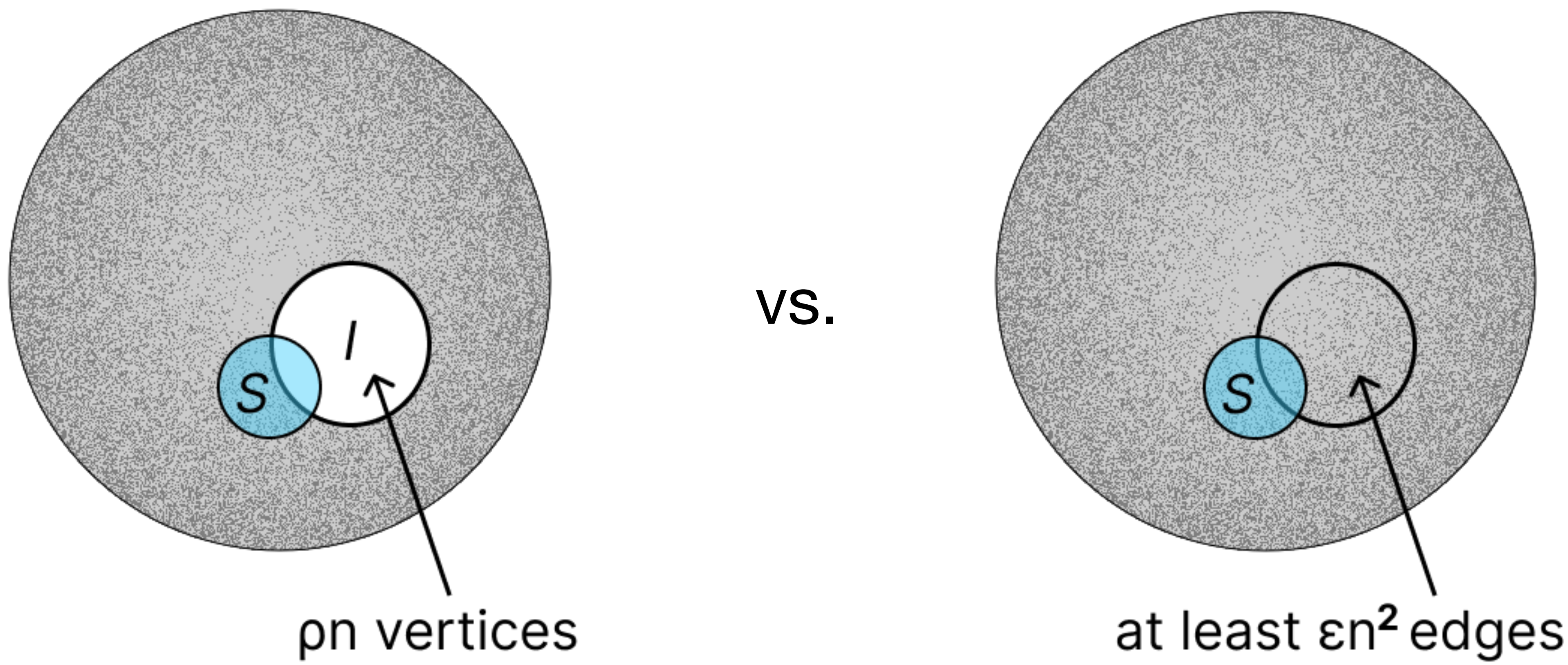
Testing Algorithm: Take a random sample S of $s \sim \rho^3/\epsilon^2$ vertices, check if the induced subgraph $G[S]$ has a ρs independent set.

Application to Testing ρ-Independent Set



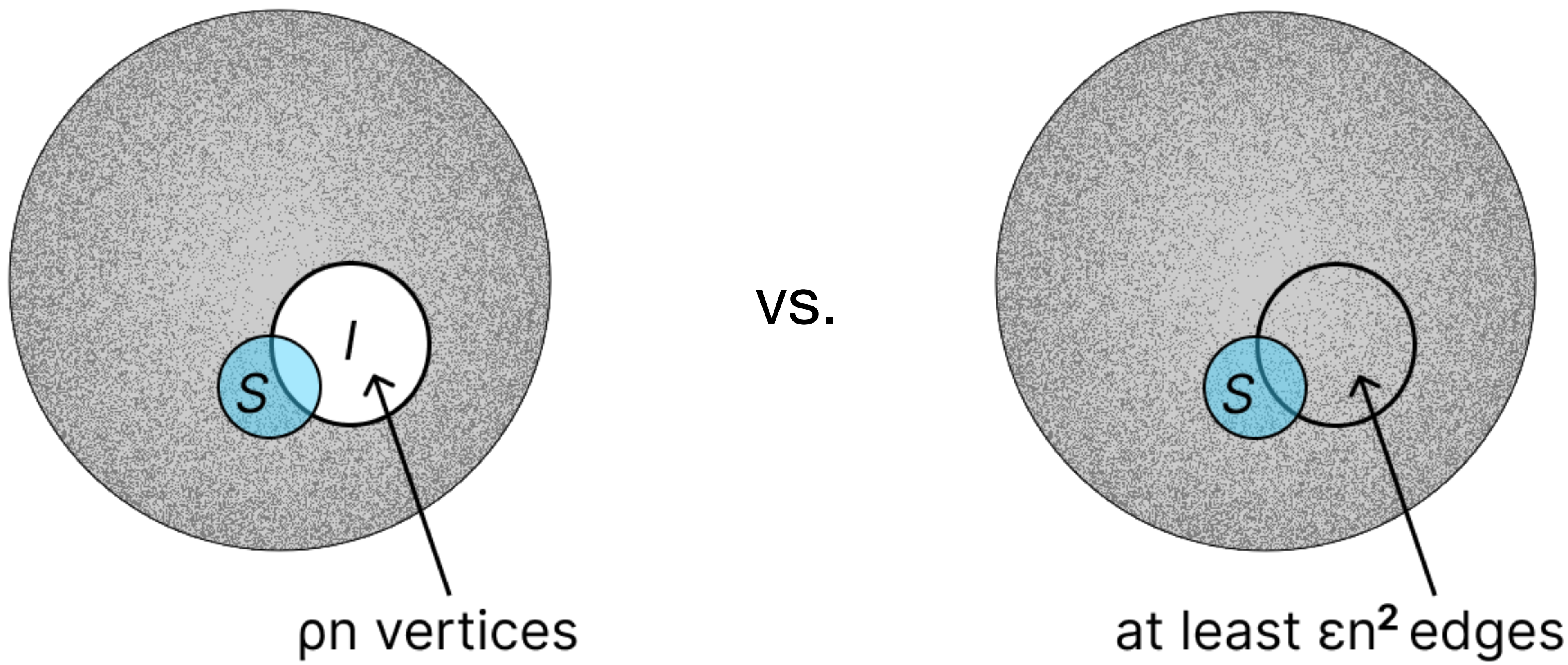
Testing Algorithm: Take a random sample S of $s \sim \rho^3/\epsilon^2$ vertices, check if the induced subgraph $G[S]$ has a ρs independent set.

Application to Testing ρ-Independent Set



Testing Algorithm: Take a random sample S of $s \sim \rho^3/\epsilon^2$ vertices, check if the induced subgraph $G[S]$ has a ρs independent set.

Application to Testing ρ-Independent Set



Testing Algorithm: Take a random sample S of $s \sim \rho^3/\epsilon^2$ vertices, check if the induced subgraph $G[S]$ has a ρs independent set.

Key Challenge: If G is ϵ-far from having a ρn independent set, show that S has a ρs independent set with only small probability.

Application of Container Lemma

Key Challenge: If G is ϵ-far from having a ρn independent set, show that S has a ρs independent set with only small probability.

Application of Container Lemma

Key Challenge: If G is ϵ-far from having a ρn independent set, show that S has a ρs independent set with only small probability.

Use Container Lemma: Every independent set is contained in a container! Each container is of size at most $|C| < (1-\epsilon)\rho n$ and there are at most $\binom{n}{1/\epsilon}$ of them.

Application of Container Lemma

Key Challenge: If G is ϵ-far from having a ρn independent set, show that S has a ρs independent set with only small probability.

Use Container Lemma: Every independent set is contained in a container! Each container is of size at most $|C| < (1-\epsilon)\rho n$ and there are at most $\binom{n}{1/\epsilon}$ of them.

$\Pr[S$ contains a ρs independent set$]$

Application of Container Lemma

Key Challenge: If G is ϵ-far from having a ρn independent set, show that S has a ρs independent set with only small probability.

Use Container Lemma: Every independent set is contained in a container! Each container is of size at most $|C| < (1-\epsilon)\rho n$ and there are at most $\binom{n}{1/\epsilon}$ of them.

$$\Pr[S \text{ contains a } \rho s \text{ independent set}]$$

$$\leq \sum_{C \in \mathscr{C}} \Pr[S \text{ contains at least } \rho s \text{ vertices from } C]$$

Application of Container Lemma

Key Challenge: If G is ϵ-far from having a ρn independent set, show that S has a ρs independent set with only small probability.

Use Container Lemma: Every independent set is contained in a container! Each container is of size at most $|C| < (1-\epsilon)\rho n$ and there are at most $\binom{n}{1/\epsilon}$ of them.

$$\Pr[S \text{ contains a } \rho s \text{ independent set}]$$

$$\leq \sum_{C \in \mathscr{C}} \Pr[S \text{ contains at least } \rho s \text{ vertices from } C]$$

$$\lesssim \binom{n}{1/\epsilon} e^{-\epsilon^2 s}$$

Application of Container Lemma

Key Challenge: If G is ϵ-far from having a ρn independent set, show that S has a ρs independent set with only small probability.

Use Container Lemma: Every independent set is contained in a container! Each container is of size at most $|C| < (1-\epsilon)\rho n$ and there are at most $\binom{n}{1/\epsilon}$ of them.

$$\Pr[S \text{ contains a } \rho s \text{ independent set}]$$

$$\leq \sum_{C \in \mathscr{C}} \Pr[S \text{ contains at least } \rho s \text{ vertices from } C]$$

$$\lesssim \binom{n}{1/\epsilon} e^{-\epsilon^2 s} \quad \leq 1/3, \text{ as long as } s \gtrsim \frac{\log n}{\epsilon^3}.$$

Application of Container Lemma

Key Challenge: If G is ϵ-far from having a ρn independent set, show that S has a ρs independent set with only small probability.

Use Container Lemma: Every independent set is contained in a container! Each container is of size at most $|C| < (1-\epsilon)\rho n$ and there are at most $\binom{n}{1/\epsilon}$ of them.

$$\Pr[S \text{ contains a } \rho s \text{ independent set}]$$

$$\leq \sum_{C \in \mathscr{C}} \Pr[S \text{ contains at least } \rho s \text{ vertices from } C]$$

$$\lesssim \binom{n}{1/\epsilon} e^{-\epsilon^2 s} \quad \leq 1/3, \text{ as long as } s \gtrsim \frac{\log n}{\epsilon^3}.$$

Takeaway:
Container lemmas imply upper bounds on the sample complexity!

Result 1: Optimal Bound on Testing the Independent Set Property

Testing Independent Sets

Theorem: There is an $\epsilon-$ tester for the ρn independent set property with sample complexity $\tilde{O}(\rho^3/\epsilon^2)$.

[Blais, Seth '23]

Remarks:

- Improves on prior best bound of $\tilde{O}(\rho^4/\epsilon^3)$
- Optimal up to log factors based on lower bound of $\Omega(\rho^3/\epsilon^2)$

[Feige, Langberg, Schechtman '04]

Testing Independent Sets: A New Container Lemma

Lemma: For any ϵ, ρ let $G = (V, E)$ be a graph such that every induced subgraph on ρn vertices has at least ϵn^2 edges. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

1. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
2. for every $C \in \mathscr{C}$, $|C| \leq (1 - \epsilon/2)\rho n$.
3. for every independent set I, there exists $C \in \mathscr{C}$ with $I \subseteq C$.

[Kleitman, Winston '82]

Testing Independent Sets: A New Container Lemma

Lemma: For any ϵ, ρ let $G = (V, E)$ be a graph such that every induced subgraph on ρn vertices has at least ϵn^2 edges. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

1. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
2. for every $C \in \mathscr{C}$, $|C| \leq (1 - \epsilon/2)\rho n$.
3. for every independent set I, there exists $C \in \mathscr{C}$ with $I \subseteq C$.

[Kleitman, Winston '82]

Question: Can there be $\binom{n}{1/\epsilon}$ containers of size $(1 - \epsilon)\rho n$?

Testing Independent Sets: A New Container Lemma

Lemma: For any ϵ, ρ let $G = (V, E)$ be a graph such that every induced subgraph on ρn vertices has at least ϵn^2 edges. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

1. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
2. for every $C \in \mathscr{C}$, $|C| \leq (1 - \epsilon/2)\rho n$.
3. for every independent set I, there exists $C \in \mathscr{C}$ with $I \subseteq C$.

[Kleitman, Winston '82]

Question: Can there be $\binom{n}{1/\epsilon}$ containers of size $(1 - \epsilon)\rho n$?

Answer: No! There are at most $\binom{n}{O(1)}$ containers of size $(1 - \epsilon)\rho n$

Testing Independent Sets: A New Container Lemma

Lemma: For any ϵ, ρ let $G = (V, E)$ be a graph such that every induced subgraph on ρn vertices has at least ϵn^2 edges. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

1. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
2. for every $C \in \mathscr{C}$, $|C| \leq (1 - \epsilon/2)\rho n$.
3. for every independent set I, there exists $C \in \mathscr{C}$ with $I \subseteq C$.

[Kleitman, Winston '82]

Question: Can there be $\binom{n}{1/\epsilon}$ containers of size $(1-\epsilon)\rho n$?

Answer: No! There are at most $\binom{n}{O(1)}$ containers of size $(1-\epsilon)\rho n$

And if there are $\binom{n}{1/\epsilon}$ containers then most of them are of size at most $(1 - \tilde{\Omega}(1))\rho n$.

Testing Independent Sets: A New Container Lemma

New Container Lemma: For any ϵ, ρ let $G = (V, E)$ be a graph such that every induced subgraph on ρn vertices has at least ϵn^2 edges. Then, there exists a set $\mathscr{F} \subseteq 2^V$ and a container function $C : \mathscr{F} \to 2^V$ that satisfies:

1. $|\mathscr{F}| \lesssim \binom{n}{1/\epsilon}$,
2. for every $F \in \mathscr{F}$, $|C(F)| \lesssim (1 - \epsilon|F|)\rho n$.
3. for every independent set I, there exists $F \in \mathscr{F}$ with $F \subseteq I \subseteq C(F)$.

Result 2: New Upper Bounds on Satisfiability Testing

Testing Satisfiability

Definition: An instance ϕ of $(q,k)-SAT$ is a constraint satisfaction problem with q variables per constraint and an alphabet size k.

ϕ is ϵ-far from satisfiable if at least $\epsilon \binom{n}{q}$ constraints must be removed to make ϕ satisfiable.

Theorem: There is an $\epsilon-$ tester $(q,k)-SAT$ property with sample complexity $\tilde{O}(kq^3/\epsilon)$.

[Blais, Seth ’24]

Testing Satisfiability

Definition: An instance ϕ of $(q, k) - SAT$ is a constraint satisfaction problem with q variables per constraint and an alphabet size k.

ϕ is ϵ-far from satisfiable if at least $\epsilon \binom{n}{q}$ constraints must be removed to make ϕ satisfiable.

Theorem: There is an $\epsilon-$ tester $(q, k) - SAT$ property with sample complexity $\tilde{O}(kq^3/\epsilon)$.

[Blais, Seth '24]

- Gives new upper bound of $\tilde{O}(k/\epsilon)$ on testing k-colorability
- Improves on prior best bounds of $2^{k^{2q}}/\epsilon^2$ and k^{3q}/ϵ

[Alon, Shapira '03] [Sohler '12]

Testing Satisfiability: New Container Lemma

Testing Algorithm: Take a random sample S of variables, accept if and only if $\phi[S]$ is satisfiable.

Testing Satisfiability: New Container Lemma

Testing Algorithm: Take a random sample S of variables, accept if and only if $\phi[S]$ is satisfiable.

Previous Container Lemmas for Independent Set Testing:

Containers for independent sets in a graph that is ϵ-far from having a ρn independent set.

New Container Lemma:

Containers for sets of variables S for which $\phi[S]$ is satisfiable in a CSP that is ϵ-far from satisfiable.

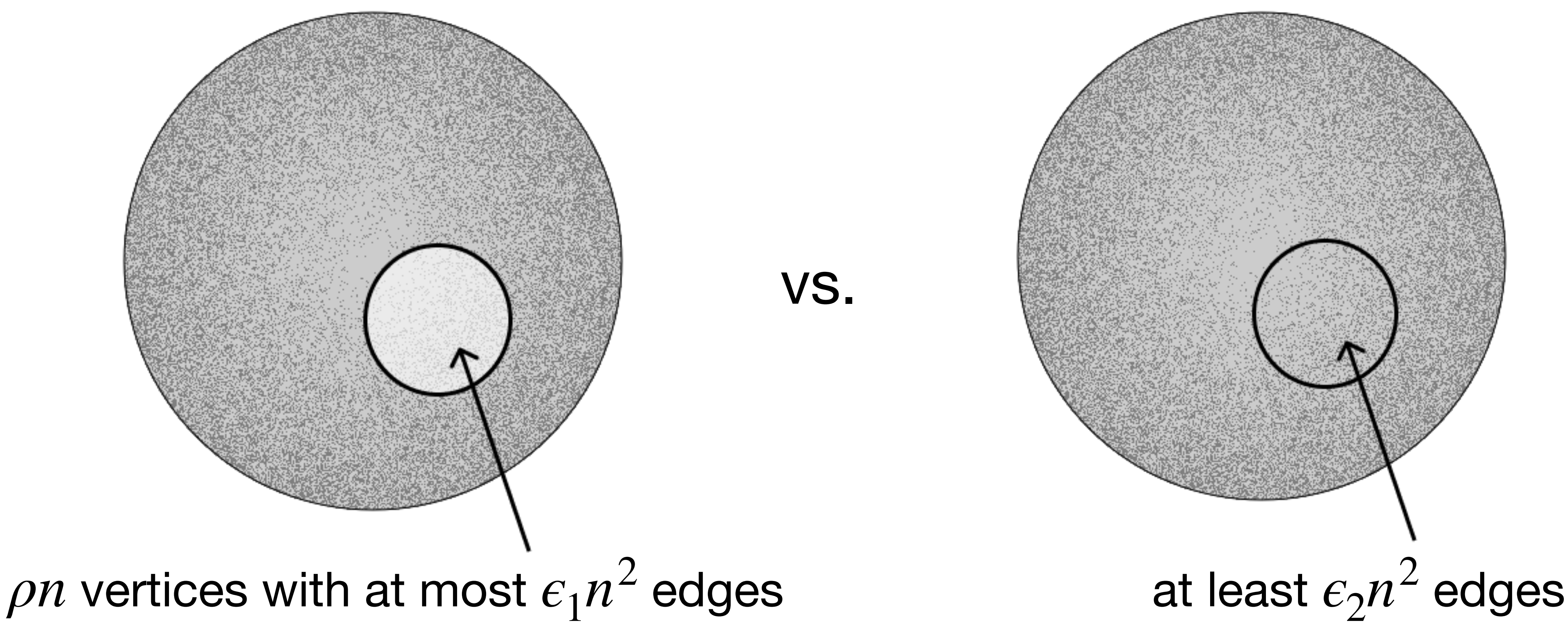
Result 3: Tolerant Testing Independent Sets

Tolerant Testing Independent Sets

Problem: For $\epsilon_1 < \epsilon_2$, distinguish between the cases:

(i) G has an induced subgraph of size ρn with fewer than $\epsilon_1 n^2$ edges (ϵ_1—close)

(ii) Every induced subgraph of size ρn has at least $\epsilon_2 n^2$ edges (ϵ_2—far)



An algorithm that, with high probability, distinguishes between (i) and (ii) is called an (ϵ_1, ϵ_2)—tester.

[Parnas, Ron, Rubinfeld '06]

Tolerant Testing Independent Sets

Theorem: There is a $\left(\frac{\epsilon}{\text{polylog}(1/\epsilon)}, \epsilon\right)-$ tolerant tester for the ρn independent set property with sample complexity $\tilde{O}(\rho^3/\epsilon^2)$.

[Seth '25]

Tolerant Testing Independent Sets

Theorem: There is a $\left(\frac{\epsilon}{\mathrm{polylog}(1/\epsilon)}, \epsilon\right)-$ tolerant tester for the ρn independent set property with sample complexity $\tilde{O}(\rho^3/\epsilon^2)$.

[Seth '25]

Remarks:

- Matches the (optimal) sample complexity bound for $\epsilon-$testing
- Best prior result is from a general result for all graph partition properties, which gives sample complexity of roughly $(1/\epsilon)^{12}$

[Fiat, Ron '21] [Goldreich, Goldwasser, Ron '98]

- In some other settings (bounded degree model, boolean strings), there is exponential gap between the query complexity of $\epsilon-$testing and $(\tilde{\Theta}(\epsilon), \epsilon)-$tolerant testing.

[Fischer, Fortnow '05] [Goldreich, Wigderson '22]

Tolerant Testing Independent Sets

Theorem: There is a $\left(\frac{\epsilon}{\text{polylog}(1/\epsilon)}, \epsilon\right)-$ tolerant tester for the ρn independent set property with sample complexity $\tilde{O}(\rho^3/\epsilon^2)$.

Tolerant Testing Independent Sets

Theorem: There is a $\left(\frac{\epsilon}{\text{polylog}(1/\epsilon)}, \epsilon\right)-$ tolerant tester for the ρn independent set property with sample complexity $\tilde{O}(\rho^3/\epsilon^2)$.

Testing Algorithm: Take a random sample S of vertices, accept if and only if $G[S]$ has a sparse induced subgraph on $\rho|S|$ vertices

New Container Lemma: Containers for *sparse subgraphs* in a graph that is ϵ-far from having a ρn independent set.

Summary of Results

	Property	Our Result	Prior Results[1]
Canonical Testers on Graphs	$\rho - \textsc{IndepSet}$	$\mathcal{S} = \tilde{O}(\rho^3/\epsilon^2)$	$\mathcal{S} = \tilde{O}(\rho^4/\epsilon^3)$, $\mathcal{S} = \Omega(\rho^3/\epsilon^2)$
	Any $k - \mathrm{SHPP}$ (for example $k - \textsc{Color}$)	$\mathcal{S} = \tilde{O}(k/\epsilon)$	$\mathcal{S} = \tilde{O}(k^6/\epsilon)$, $\mathcal{S} = \tilde{O}(k/\epsilon^2)$, $\mathcal{S} = \tilde{\Omega}(1/\epsilon)$
Other Canonical Testers	$(q,k) - \textsc{Sat}$	$\mathcal{S} = \tilde{O}(q^3 k/\epsilon)$	$\mathcal{S} = (kq)^{O(q)}/\epsilon$
Tolerant Tester	$\rho - \textsc{IndepSet}$	$\mathcal{S} = \tilde{O}(\rho^3/\epsilon^2)$	$\mathcal{S} = \tilde{O}(1/\epsilon^{12})$, $\mathcal{S} = \Omega(\rho^3/\epsilon^2)$
Non-Canonical Tester	$\rho - \textsc{IndepSet}$	$\mathcal{Q} = \tilde{O}(\rho^4/\epsilon^3)$	$\mathcal{Q} = \tilde{O}(\mathcal{S}^2)$, $\mathcal{Q} = \Omega(1/\epsilon)$

[1] Prior results by [Feige, Langberg, Schechtman '04], [Sohler '12], [Alon, Krivelevich '02] and [Fiat, Ron '21]

Summary of Results

	Property	Our Result	Prior Results[1]
Canonical Testers on Graphs	$\rho - \textsc{IndepSet}$	$\mathcal{S} = \tilde{O}(\rho^3/\epsilon^2)$	$\mathcal{S} = \tilde{O}(\rho^4/\epsilon^3)$, $\mathcal{S} = \Omega(\rho^3/\epsilon^2)$
	Any $k - \mathrm{SHPP}$ (for example $k - \textsc{Color}$)	$\mathcal{S} = \tilde{O}(k/\epsilon)$	$\mathcal{S} = \tilde{O}(k^6/\epsilon)$, $\mathcal{S} = \tilde{O}(k/\epsilon^2)$, $\mathcal{S} = \tilde{\Omega}(1/\epsilon)$
Other Canonical Testers	$(q, k) - \textsc{Sat}$	$\mathcal{S} = \tilde{O}(q^3 k/\epsilon)$	$\mathcal{S} = (kq)^{O(q)}/\epsilon$
Tolerant Tester	$\rho - \textsc{IndepSet}$	$\mathcal{S} = \tilde{O}(\rho^3/\epsilon^2)$	$\mathcal{S} = \tilde{O}(1/\epsilon^{12})$, $\mathcal{S} = \Omega(\rho^3/\epsilon^2)$
Non-Canonical Tester	$\rho - \textsc{IndepSet}$	$\mathcal{Q} = \tilde{O}(\rho^4/\epsilon^3)$	$\mathcal{Q} = \tilde{O}(\mathcal{S}^2)$, $\mathcal{Q} = \Omega(1/\epsilon)$

[1] Prior results by [Feige, Langberg, Schechtman '04], [Sohler '12], [Alon, Krivelevich '02] and [Fiat, Ron '21]

Thank you!

How to Remove $\log n$ Factor

Basic Application of Container Lemma

Key Challenge: If G is ϵ-far from having a ρn independent set, show that S has a ρs independent set with only small probability.

Use Container Lemma: Every independent set is contained in a container! Each container is of size at most $|C| < (1-\epsilon)\rho n$ and there are at most $\binom{n}{1/\epsilon}$ of them.

$\Pr[S$ contains a ρs independent set$]$

$\leq \sum_{C \in \mathscr{C}} \Pr[S$ contains at least ρs vertices from $C]$

$\lesssim \binom{n}{1/\epsilon} e^{-\epsilon^2 s} \quad \leq 1/3$, as long as $s \gtrsim \dfrac{\log n}{\epsilon^3}$.

Better Application of Container Lemma

Key Challenge: If G is ϵ-far from having a ρn independent set, show that S has a ρs independent set with only small probability.

Use Container Lemma (with fingerprints): There is a collection $\mathcal{F}$ of fingerprints such that for every I there exists $F \in \mathcal{F}$ with $F \subseteq I \subseteq C(F)$.

$$\Pr[S \text{ contains a } \rho s \text{ independent set}]$$

$$\leq \sum_{F \in \mathcal{F}} \Pr[F \subseteq S \text{ and } S \text{ contains at least } \rho s \text{ vertices from } C(F)]$$

$$\leq \sum_{F \in \mathcal{F}} \Pr[F \subseteq S] \cdot \Pr[S \text{ contains at least } \rho s \text{ vertices from } C(F) \mid F \subseteq S]$$

$$\frac{\binom{n-|F|}{|S|-|F|}}{\binom{n}{|S|}} = \frac{\binom{s}{|F|}}{\binom{n}{|F|}}$$

How to prove new Container Lemma for Sparse Sets - An Encoding Argument

A Container Lemma For Sparse Sets (restated)

Lemma*: For any ϵ, ρ let $G = (V, E)$ be a graph such that every induced subgraph on ρn vertices has at least ϵn^2 edges. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

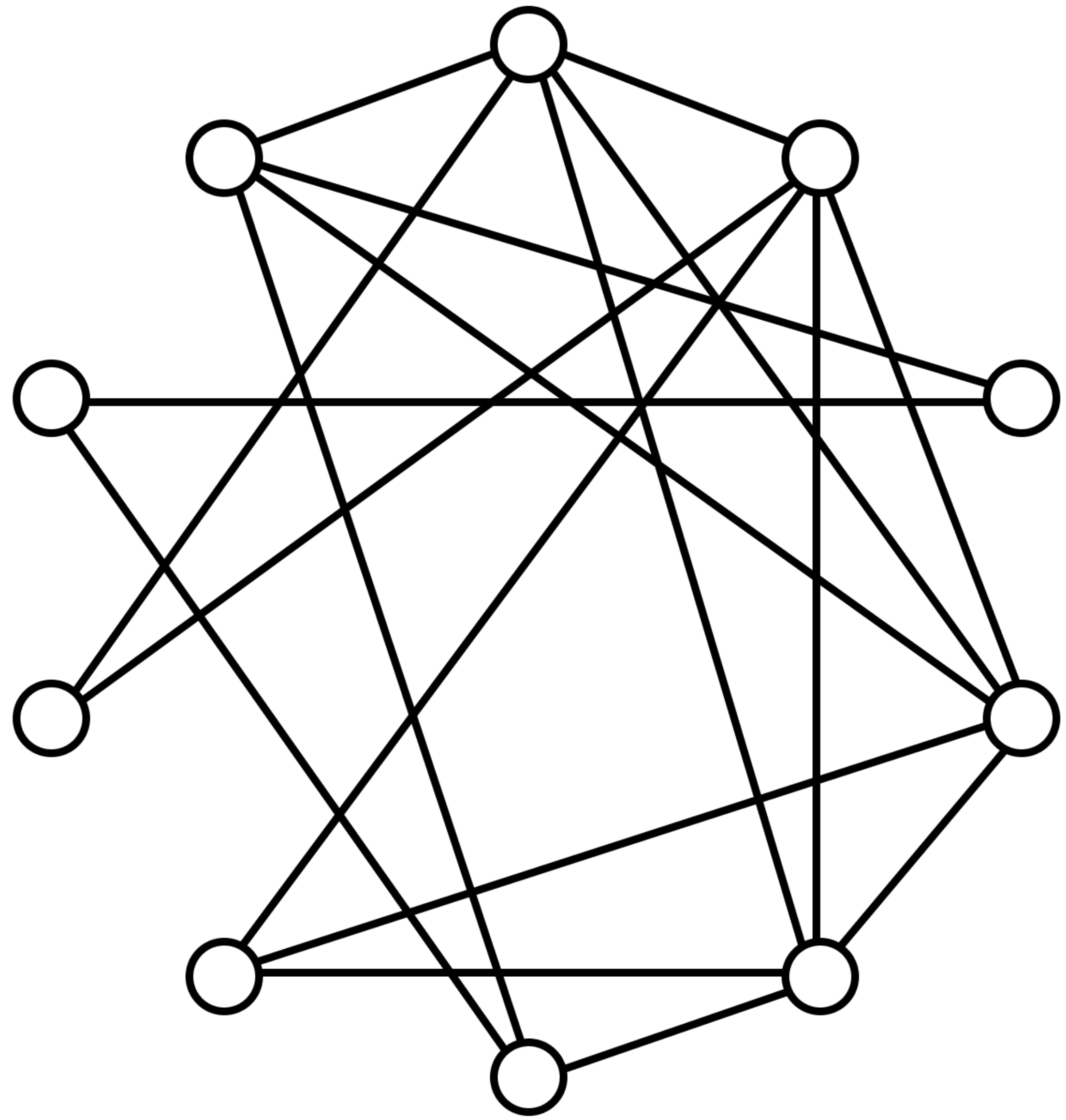
1. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
2. for every $C \in \mathscr{C}$, $|C| < (1-\epsilon)\rho n$.
3. For every set $J \subseteq V$ such that $G[J]$ has fewer than $\frac{\epsilon}{\text{polylog}(1/\epsilon)} |J|^2$ edges, there exists $C \in \mathscr{C}$ and α such that $|C| = (1-\alpha)\rho n$ and
$$|C \cap J| \geq \left(1 - \frac{\alpha}{2}\right)|J|.$$

Informally: each sparse set is "mostly" contained in a container

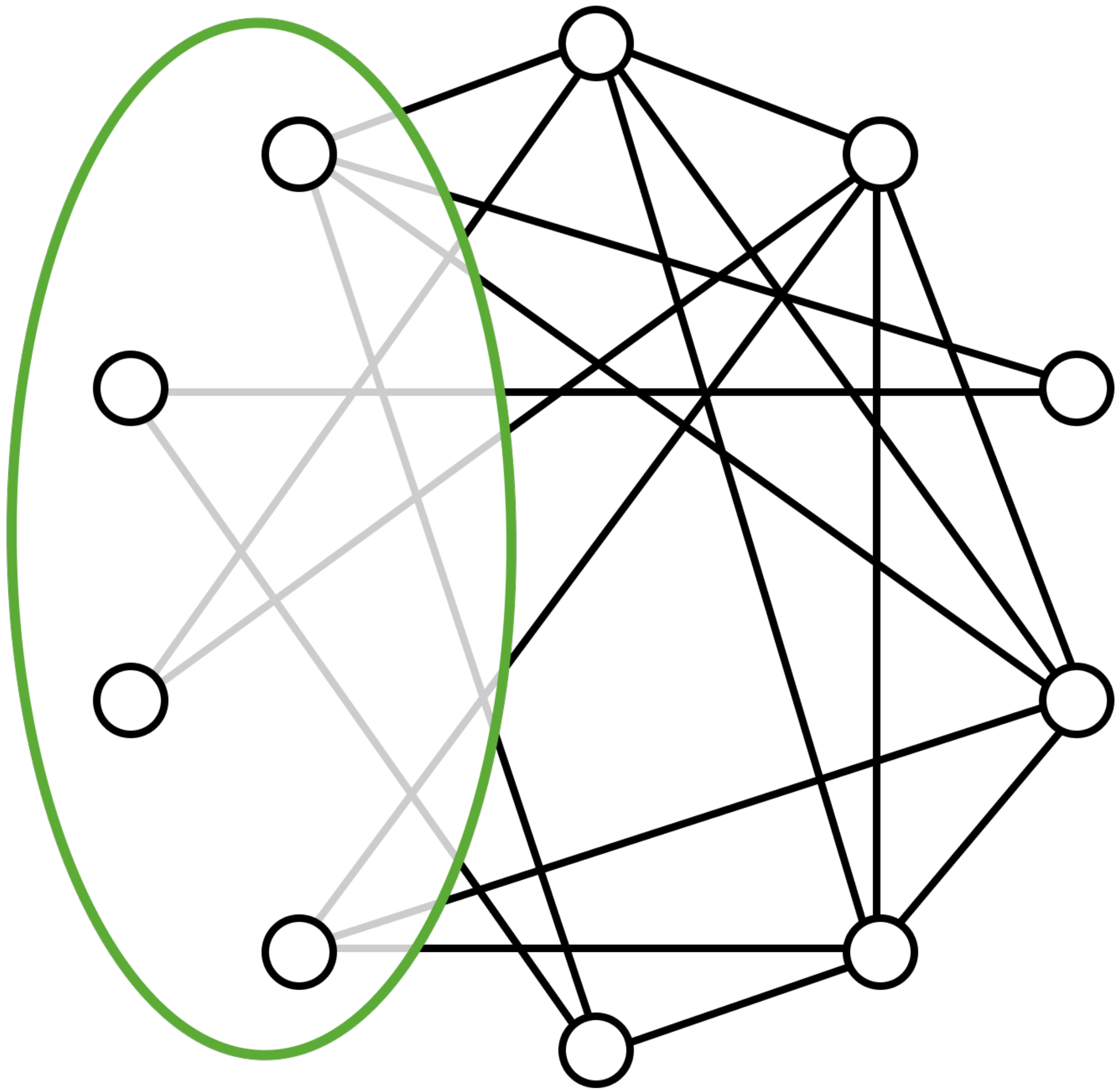
* Omitting some details

Encoding Sparse Sets

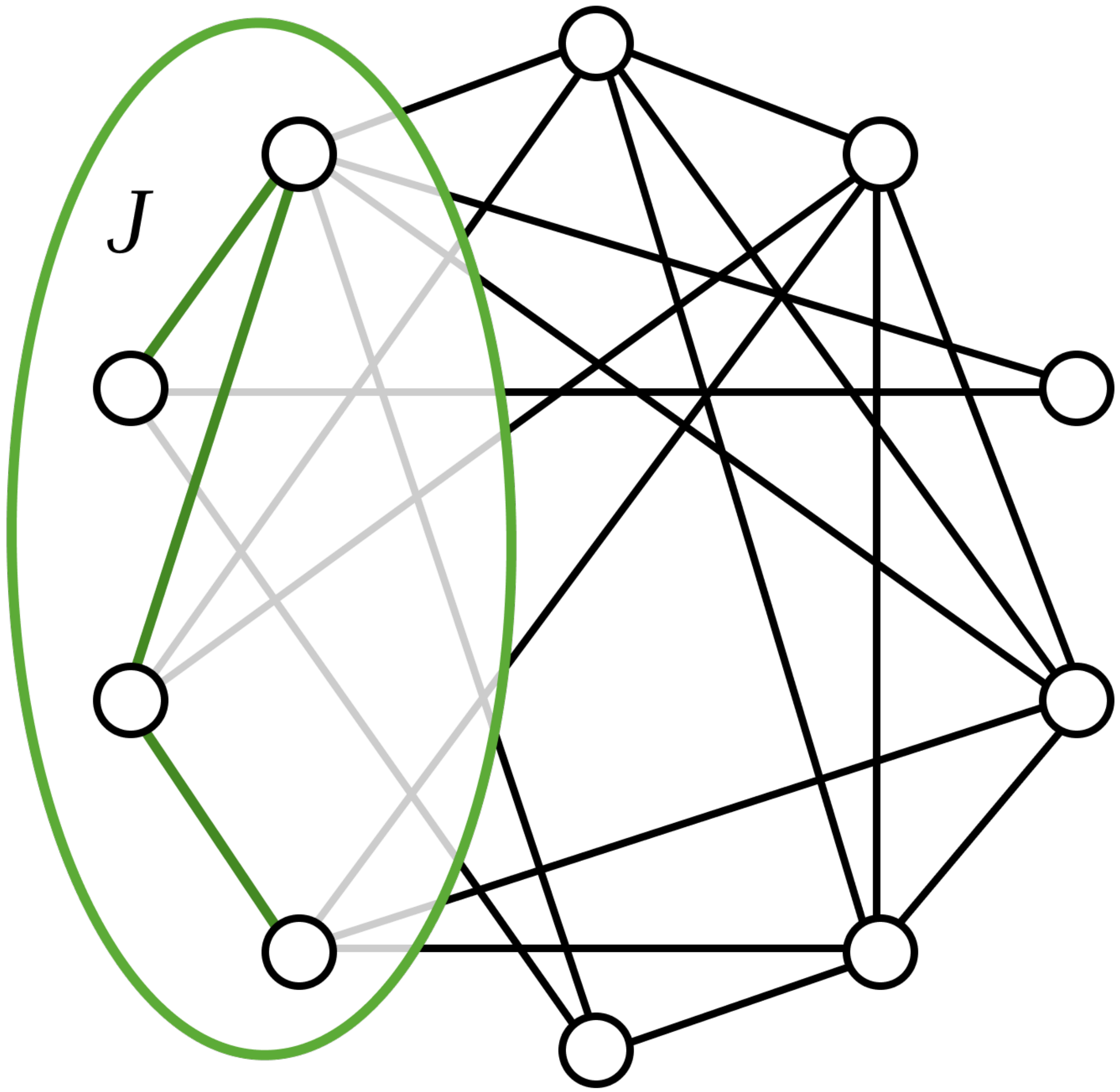
Encoding Sparse Sets



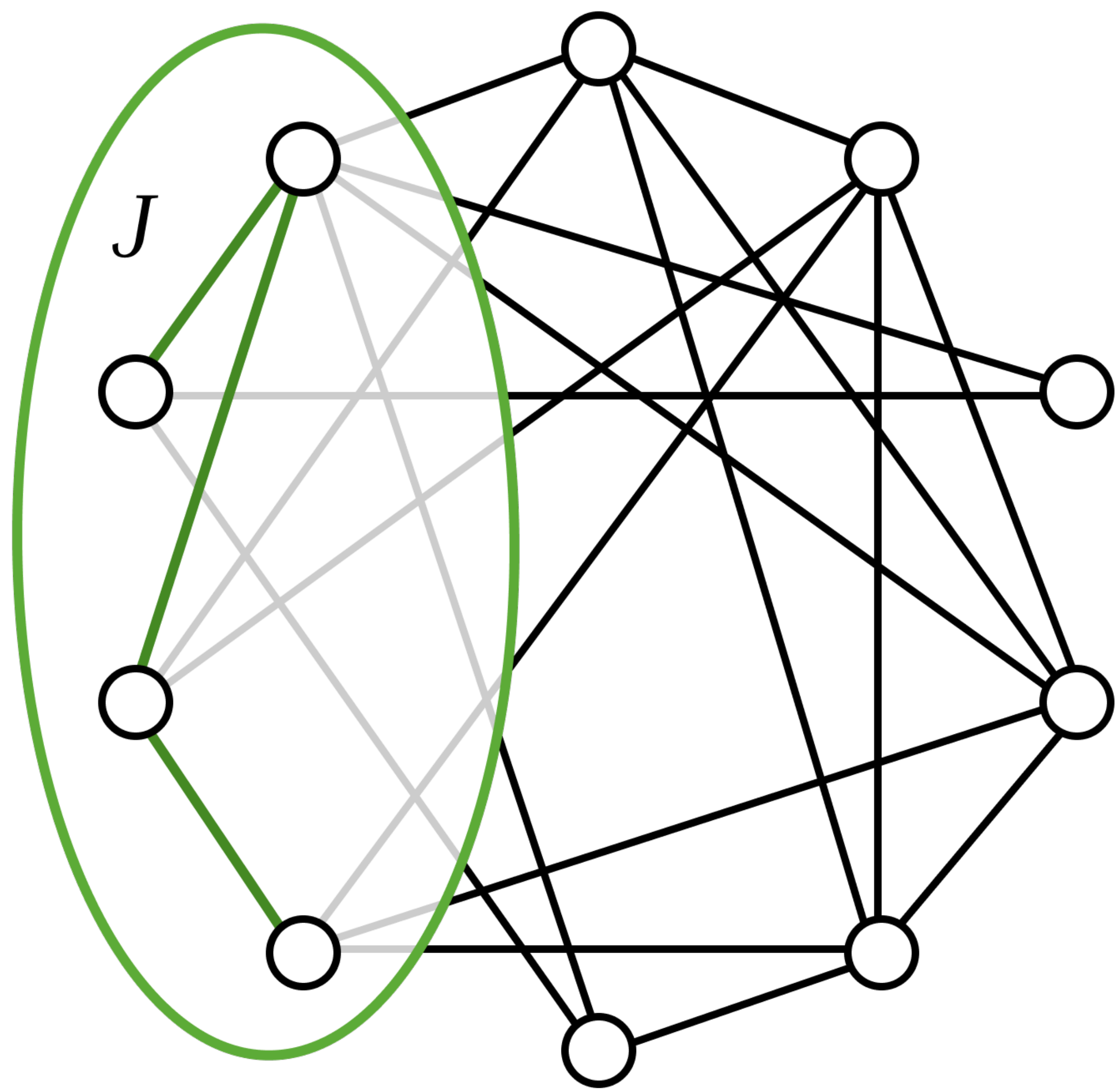
Encoding Sparse Sets



Encoding Sparse Sets

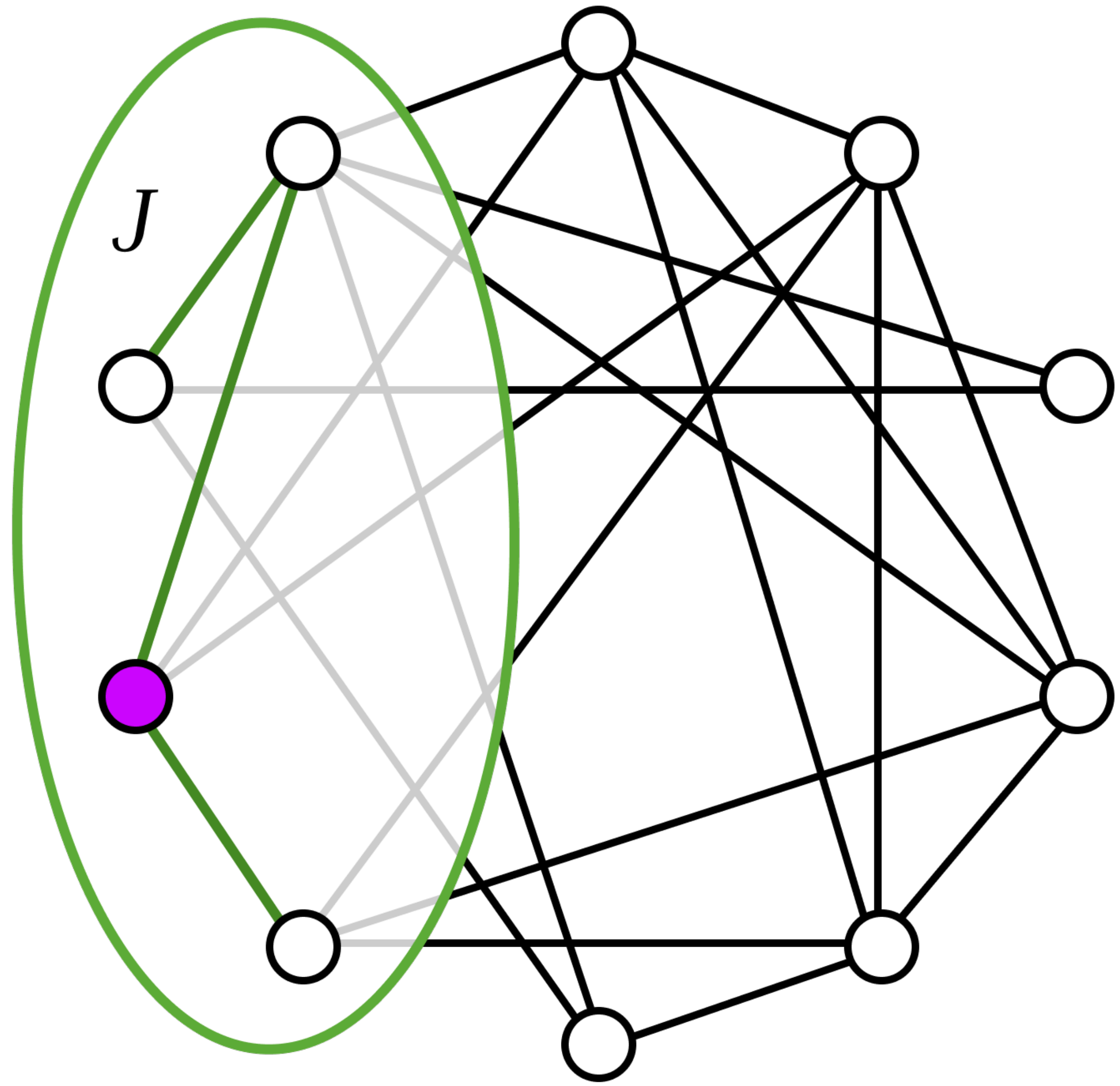


Encoding Sparse Sets

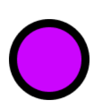


How can I give you the most information about a sparse set J by just telling you about one (or two) of the vertices in J?

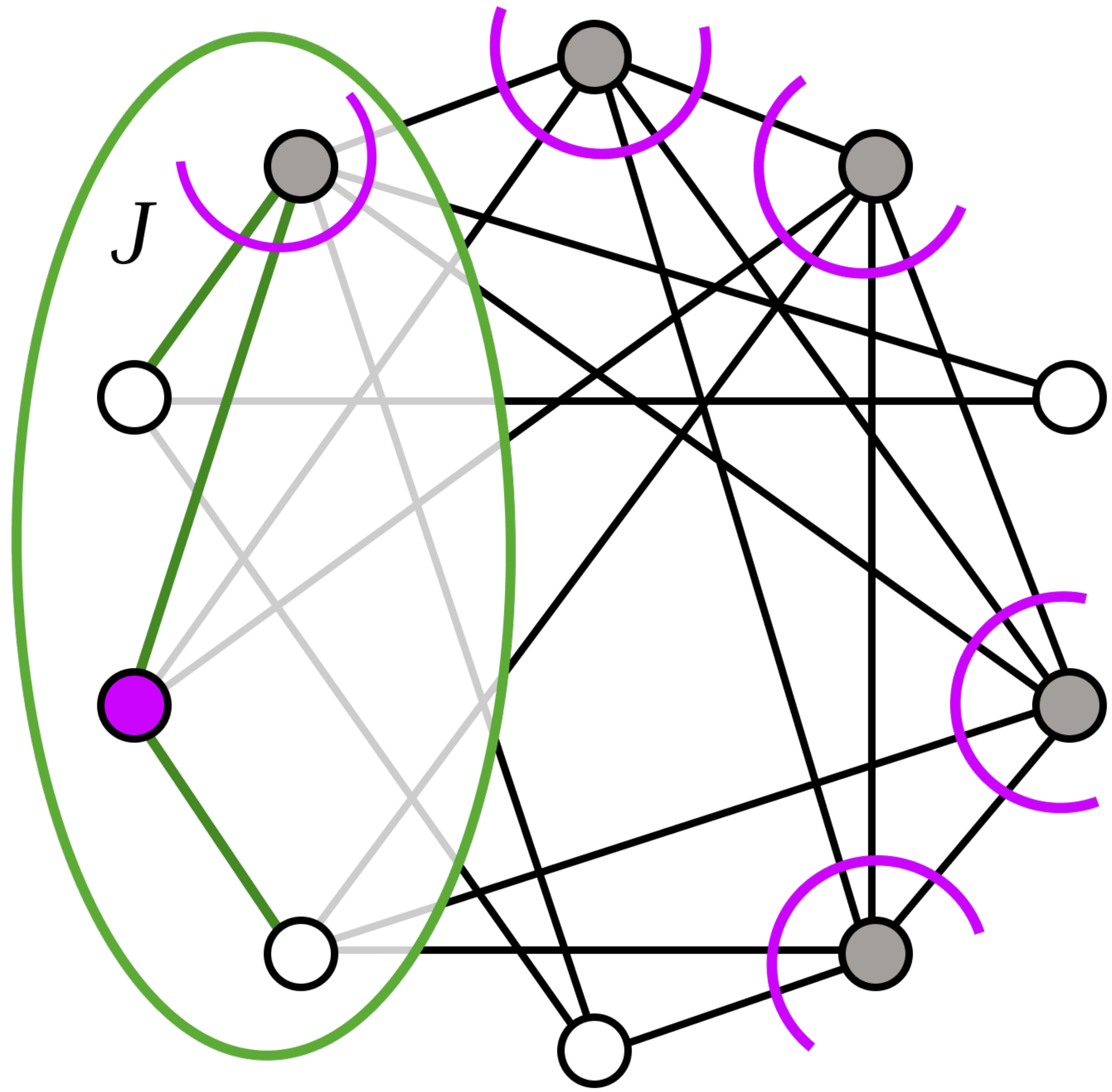
Encoding Sparse Sets



How can I give you the most information about a sparse set J by just telling you about one (or two) of the vertices in J?

Answer: Send  and remove higher degree vertices

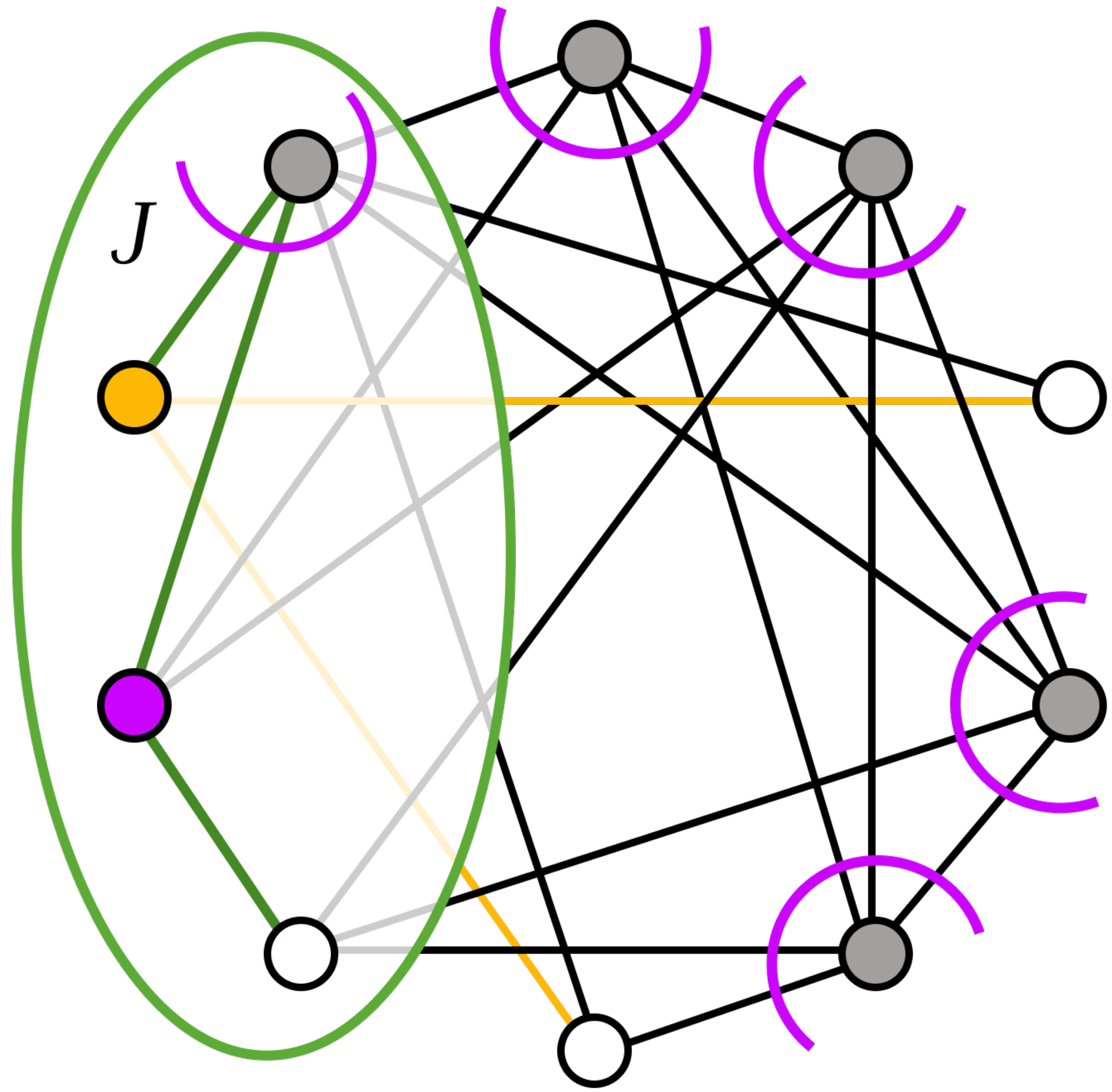
Encoding Sparse Sets



How can I give you the most information about a sparse set J by just telling you about one (or two) of the vertices in J?

Answer: Send and remove higher degree vertices

Encoding Sparse Sets

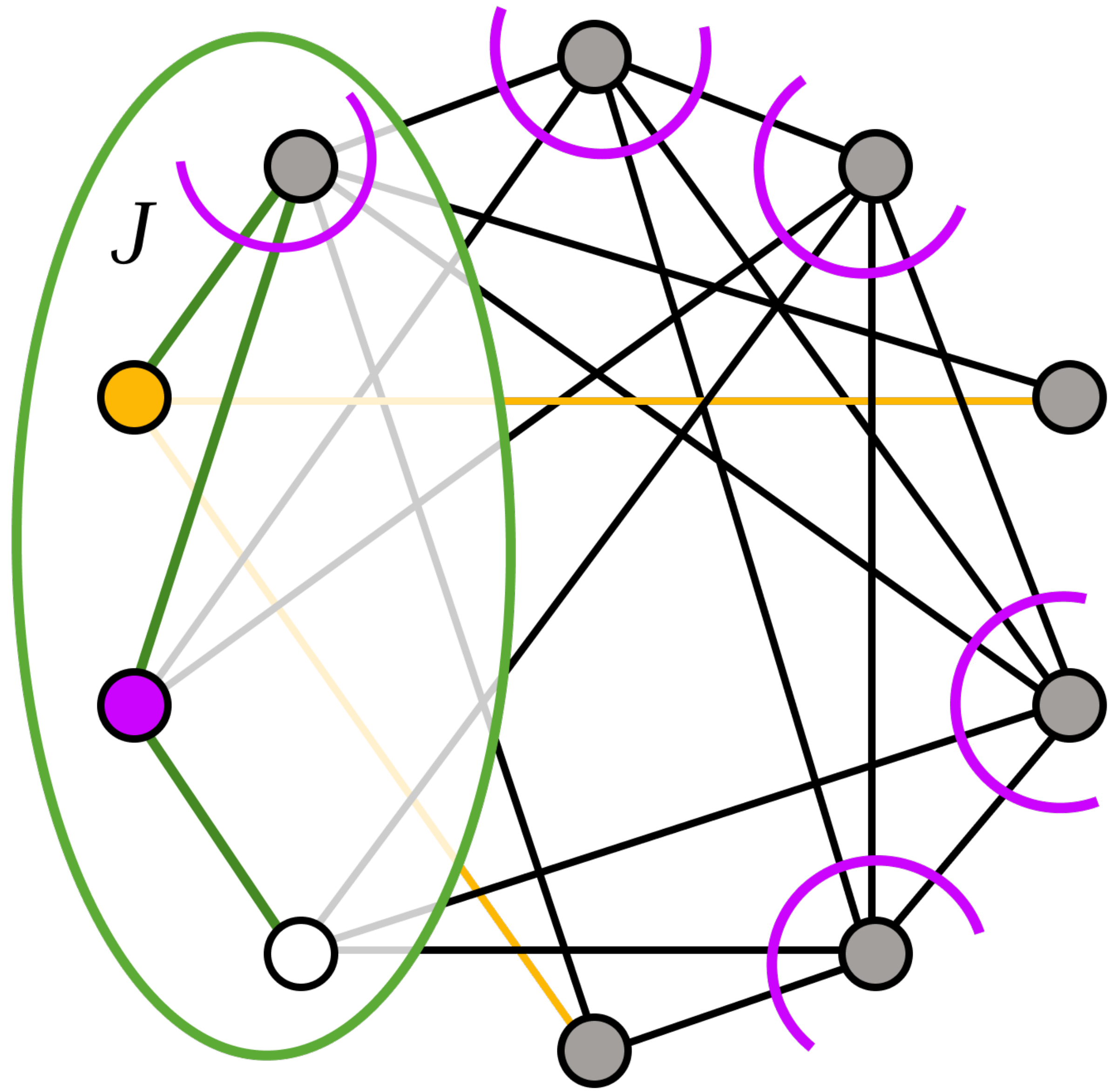


How can I give you the most information about a sparse set J by just telling you about one (or two) of the vertices in J?

Answer: Send ● and remove higher degree vertices

Send ● and remove neighbours

Encoding Sparse Sets



How can I give you the most information about a sparse set J by just telling you about one (or two) of the vertices in J?

Answer: Send and remove higher degree vertices

Send and remove neighbours

Container Generator Algorithm - Sparse Sets

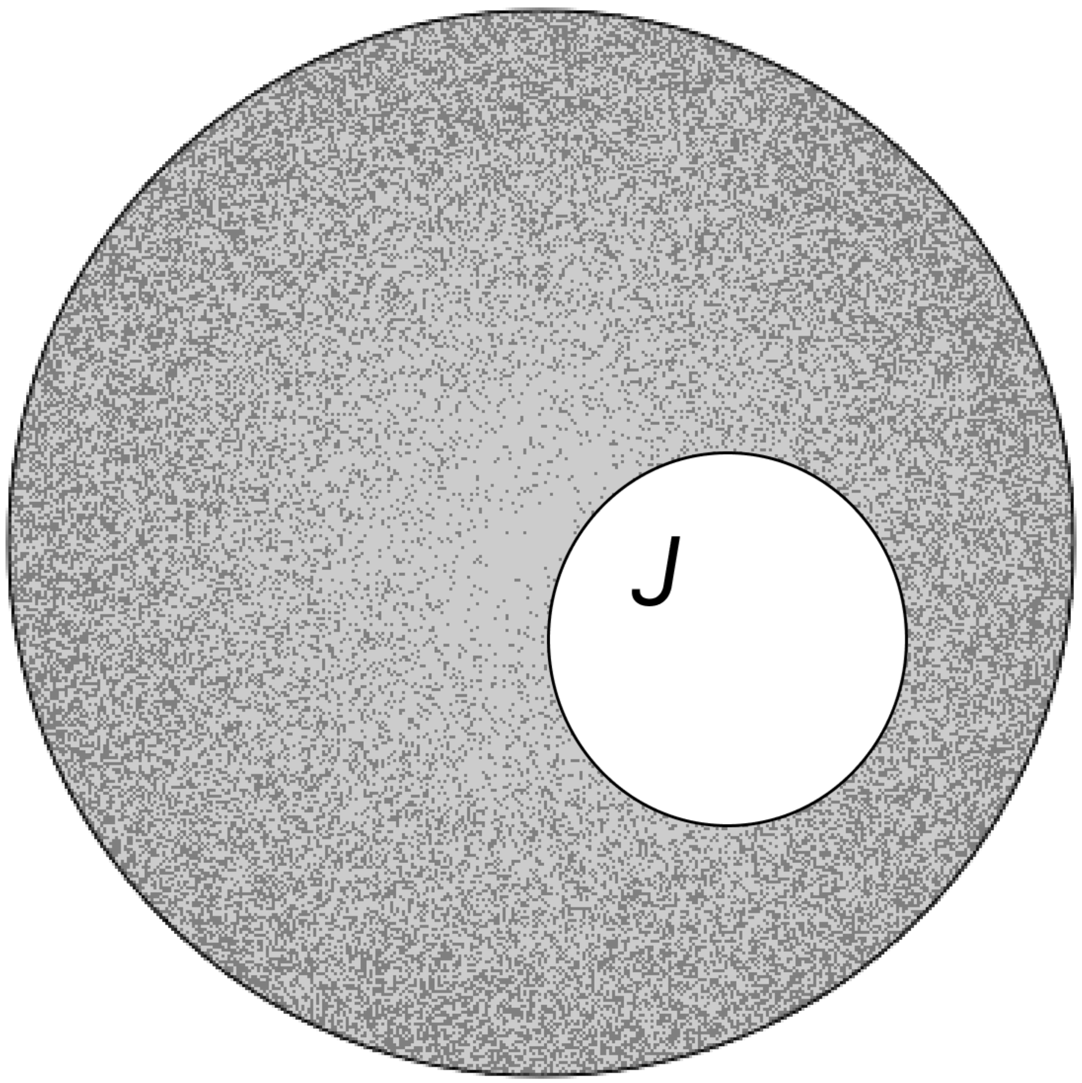
Input: Graph G and a sparse set J

- Initialize fingerprint $F = \emptyset$ and container $C = V$
- Repeat while $G[C]$ has more than ϵn^2 edges
  - select a vertex $v \in J$ and an operation of "remove neighbours" or "remove higher degree vertices" that maximizes the following:

$$\frac{\text{\# vertices removed from } C \text{ by operation}}{\text{\# vertices in } J \text{ removed from } C \text{ by operation}}$$

  - apply operation to container, add v and operation to F

Container Generator Algorithm - Sparse Sets



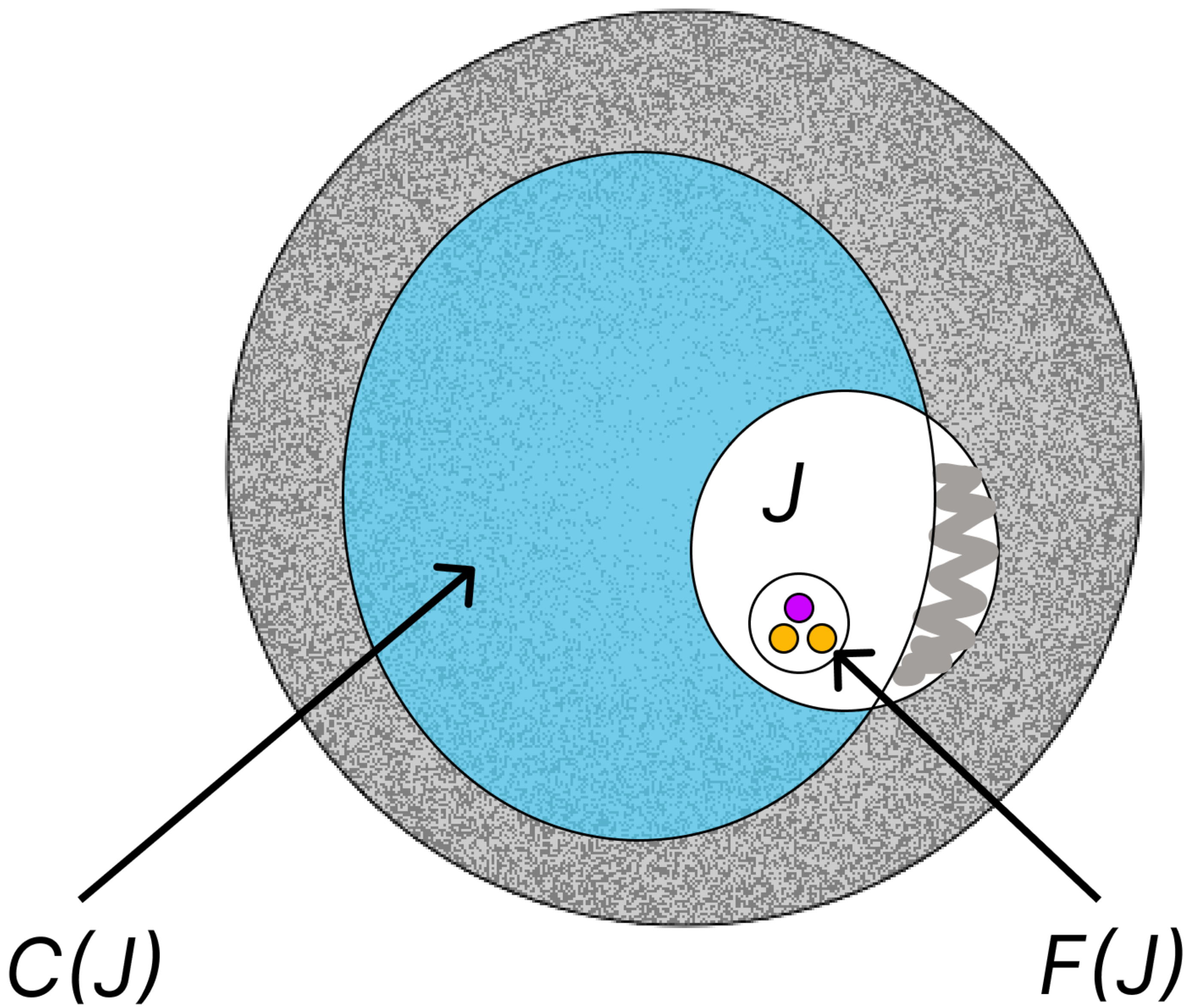
Input: Graph G and a sparse set J

- Initialize fingerprint $F = \varnothing$ and container $C = V$
- Repeat while $G[C]$ has more than ϵn^2 edges
  - select a vertex $v \in J$ and an operation of "remove neighbours" or "remove higher degree vertices" that maximizes the following:

$$\frac{\text{\# vertices removed from } C \text{ by operation}}{\text{\# vertices in } J \text{ removed from } C \text{ by operation}}$$

  - apply operation to container, add v and operation to F

Container Generator Algorithm - Sparse Sets



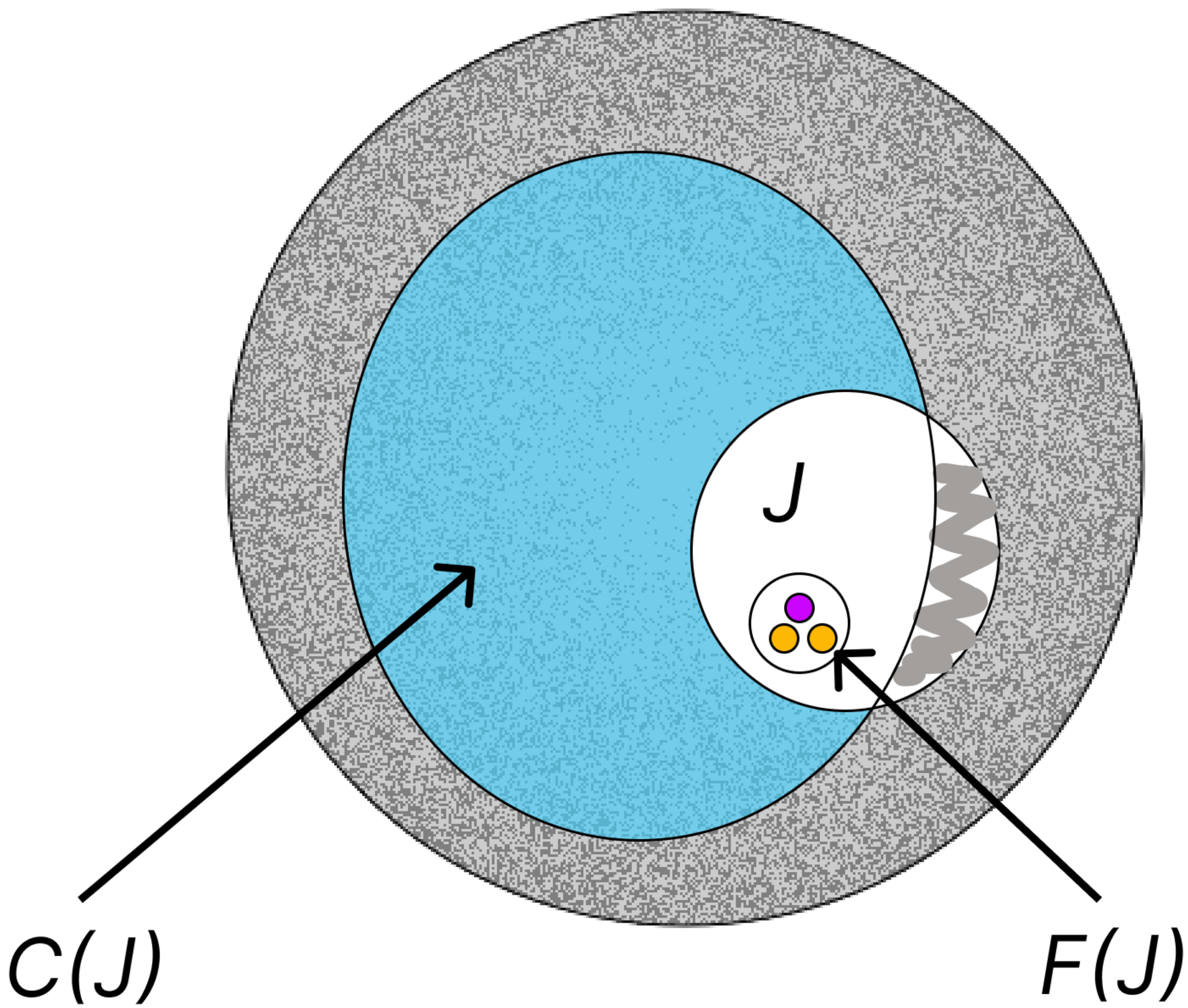
Input: Graph G and a sparse set J

- Initialize fingerprint $F = \emptyset$ and container $C = V$
- Repeat while $G[C]$ has more than ϵn^2 edges
  - select a vertex $v \in J$ and an operation of "remove neighbours" or "remove higher degree vertices" that maximizes the following:

$$\frac{\text{\# vertices removed from } C \text{ by operation}}{\text{\# vertices in } J \text{ removed from } C \text{ by operation}}$$

  - apply operation to container, add v and operation to F

Container Generator Algorithm - Sparse Sets



Input: Graph G and a sparse set J

- Initialize fingerprint $F = \emptyset$ and container $C = V$
- Repeat while $G[C]$ has more than ϵn^2 edges
  - select a vertex $v \in J$ and an operation of "remove neighbours" or "remove higher degree vertices" that maximizes the following:

$$\frac{\text{\# vertices removed from } C \text{ by operation}}{\text{\# vertices in } J \text{ removed from } C \text{ by operation}}$$

  - apply operation to container, add v and operation to F

Observations:

- $C(J)$ can be reconstructed from $F(J)$
- J is NOT fully contained in $C(J)$