Linear-Time Counting Algorithms for Independent Sets in Chordal Graphs

Yoshio Okamoto	Toyohashi Univ Tech	JP
Takeaki Uno	National Inst Informatics	JP
Ryuhei Uehara	Japan Adv Inst Sci Tech	JP

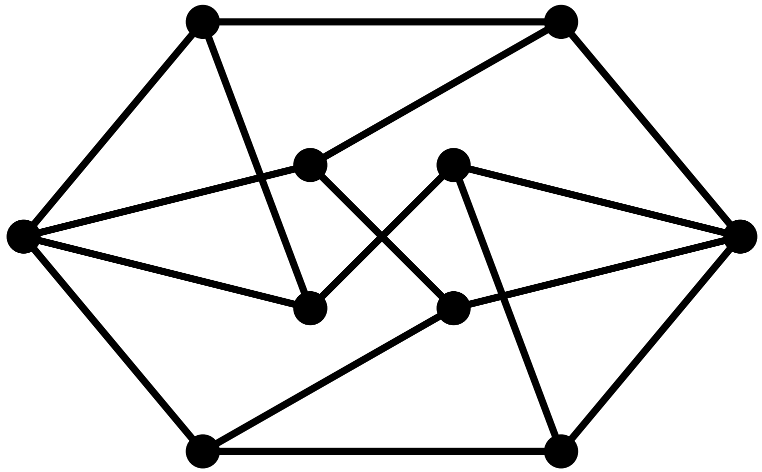
June 25, 2005

31st International Workshop on Graph-Theoretic Concepts in Computer Science

University of Metz, Metz, France

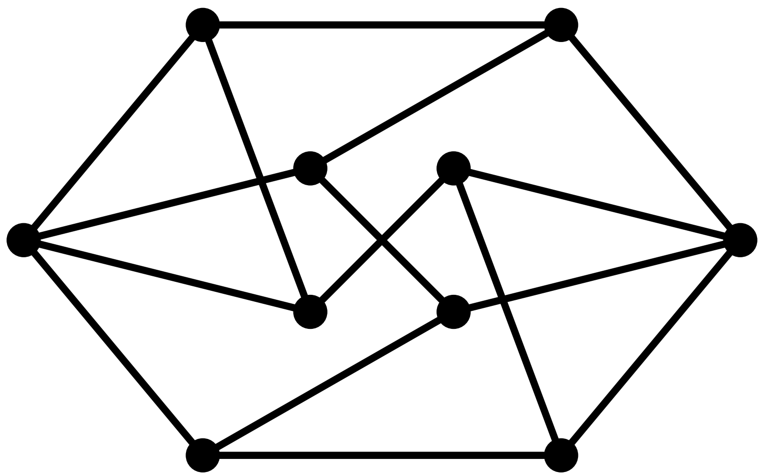
$\boxed{\text{Setup}}$ $G = (V, E)$ a graph (undirected, finite, simple)

Setup $G = (V, E)$ a graph (undirected, finite, simple)



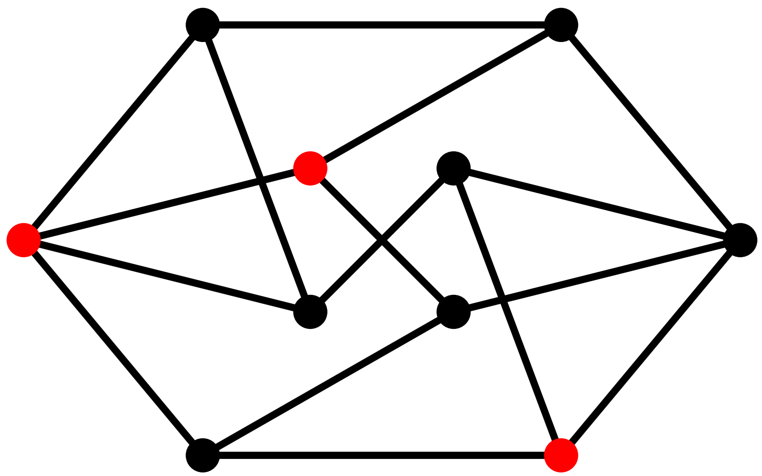
Setup $G = (V, E)$ a graph (undirected, finite, simple)

Def. A set $I \subseteq V$ is an **independent set** of G if no two vertices in I are adjacent.



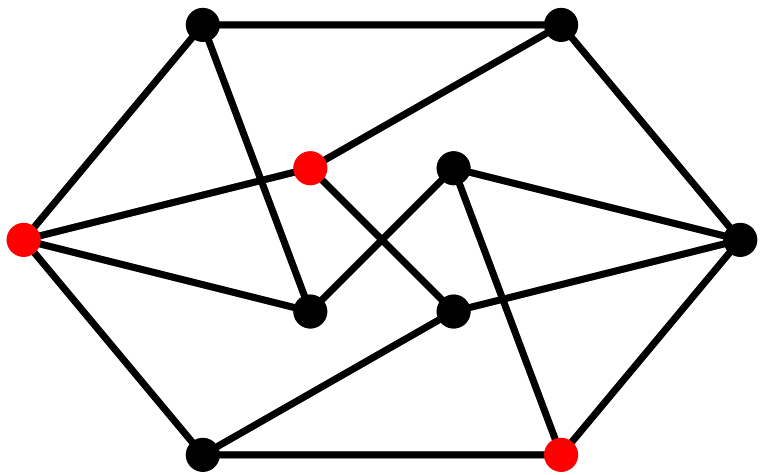
Setup $G = (V, E)$ a graph (undirected, finite, simple)

Def. A set $I \subseteq V$ is an **independent set** of G if no two vertices in I are adjacent.



Setup $G = (V, E)$ a graph (undirected, finite, simple)

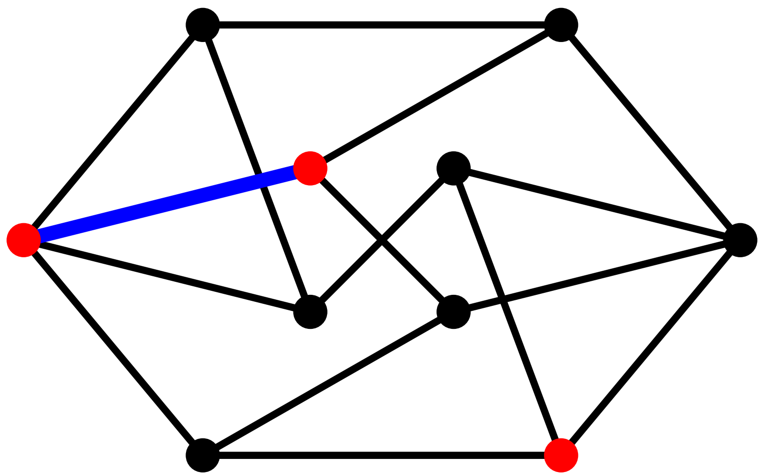
Def. A set $I \subseteq V$ is an **independent set** of G if no two vertices in I are adjacent.



Not an independent set

Setup $G = (V, E)$ a graph (undirected, finite, simple)

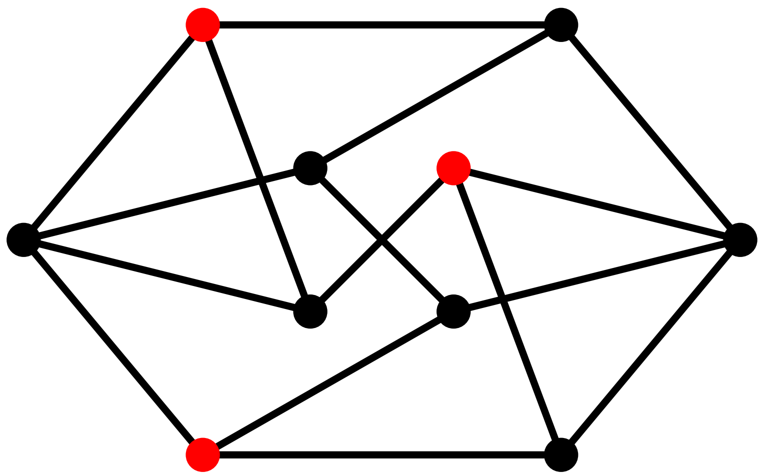
Def. A set $I \subseteq V$ is an **independent set** of G if no two vertices in I are adjacent.



Not an independent set

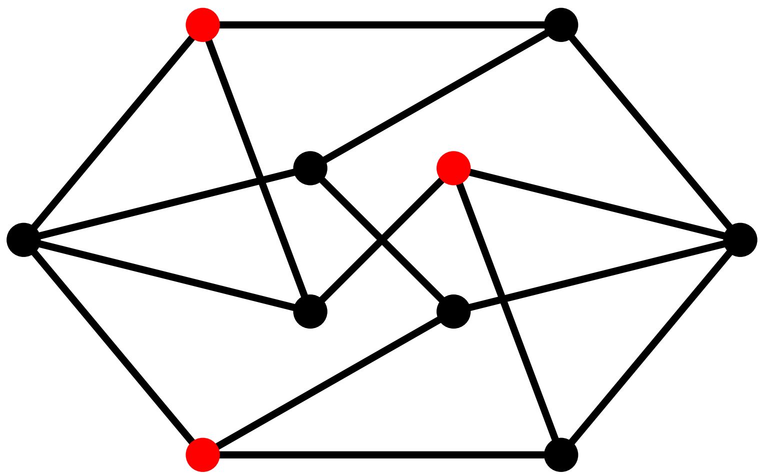
Setup $G = (V, E)$ a graph (undirected, finite, simple)

Def. A set $I \subseteq V$ is an **independent set** of G if no two vertices in I are adjacent.



Setup $G = (V, E)$ a graph (undirected, finite, simple)

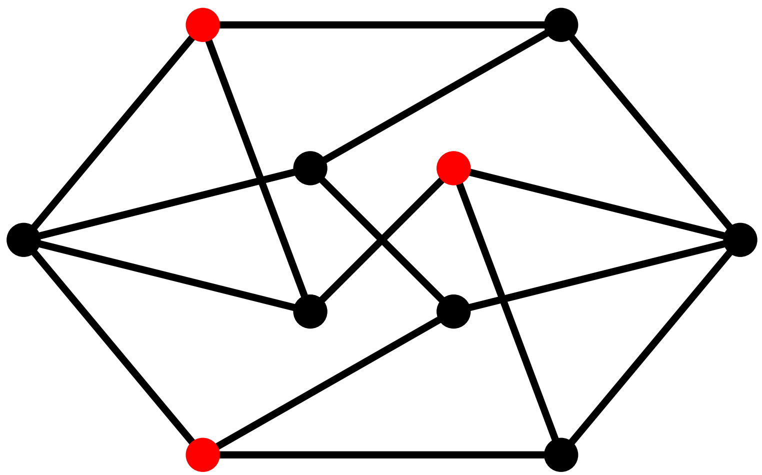
Def. A set $I \subseteq V$ is an **independent set** of G if no two vertices in I are adjacent.



An independent set

Setup $G = (V, E)$ a graph (undirected, finite, simple)

Def. A set $I \subseteq V$ is an **independent set** of G if no two vertices in I are adjacent.

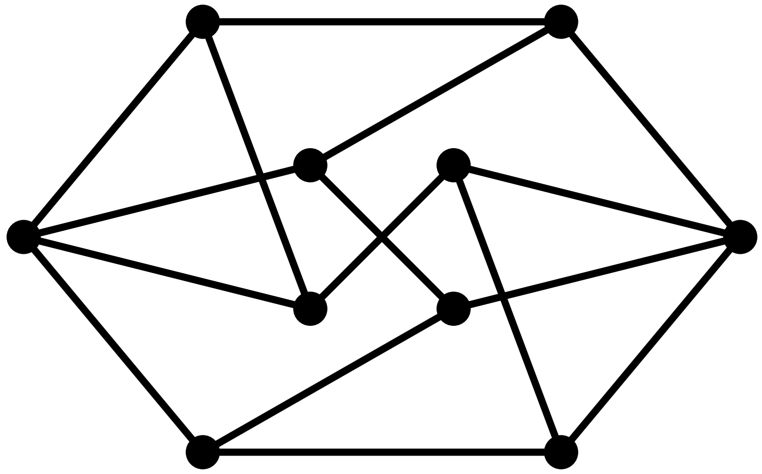


An independent set

Also called a stable set of G

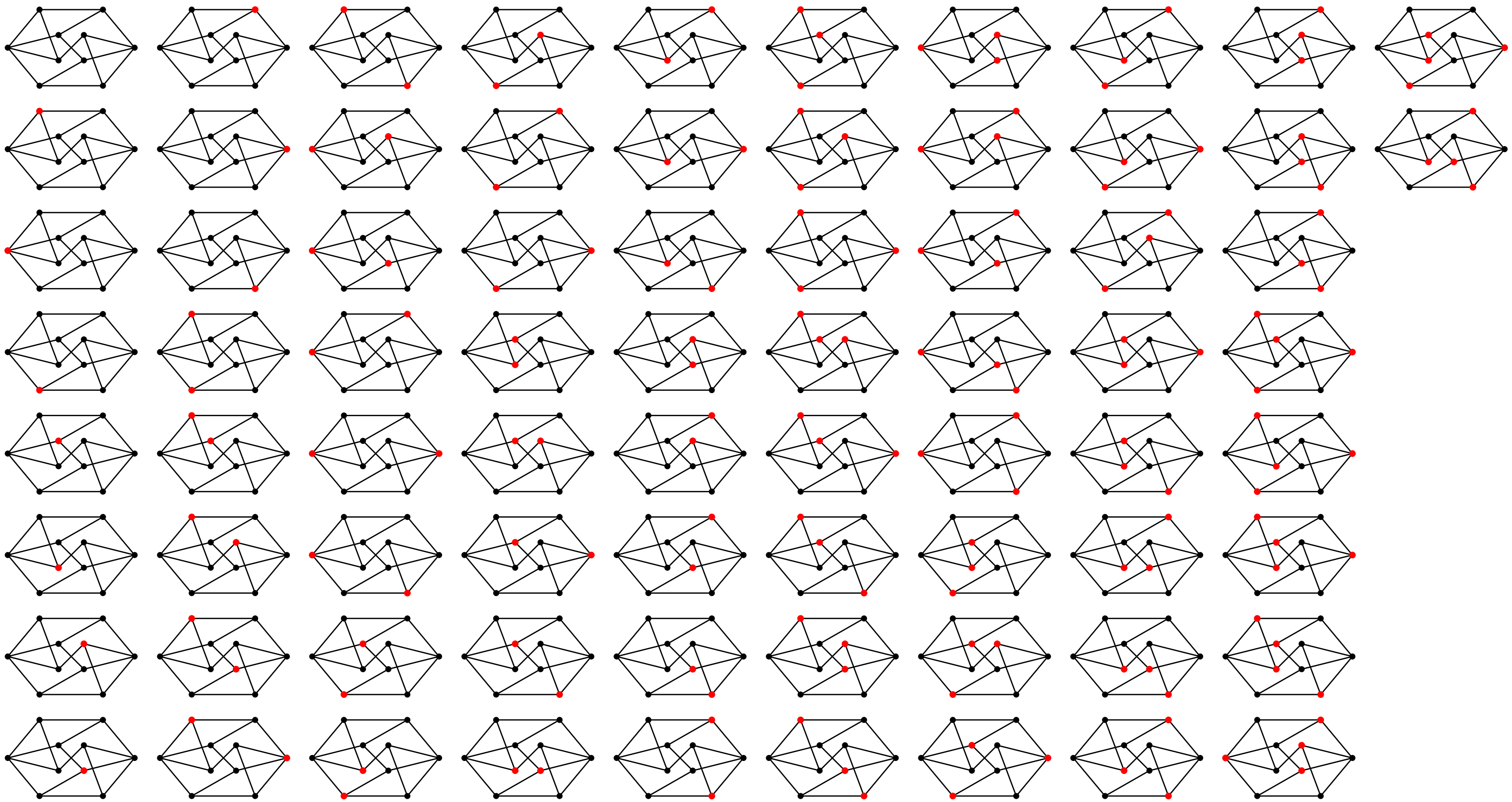
$\boxed{\text{Input}}$ $G = (V, E)$ a graph

$\boxed{\text{Output}}$ All independent sets of G



Input $G = (V, E)$ a graph

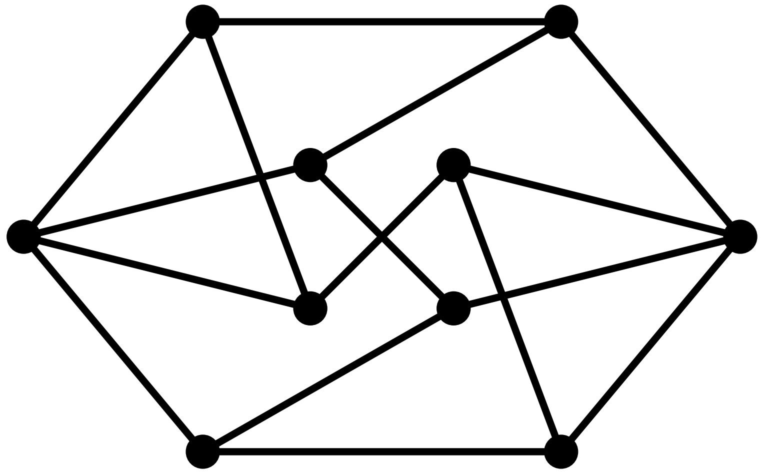
Output All independent sets of G



74 independent sets in total

Input $G = (V, E)$ a graph

Output # independent sets of G



Input $G = (V, E)$ a graph

Output # independent sets of G

74

We study the following **counting problems**:

Input $G = (V, E)$ a graph

Output (1) # independent sets of G

(2) # maximum independent sets of G

(3) # independent sets of G of fixed size

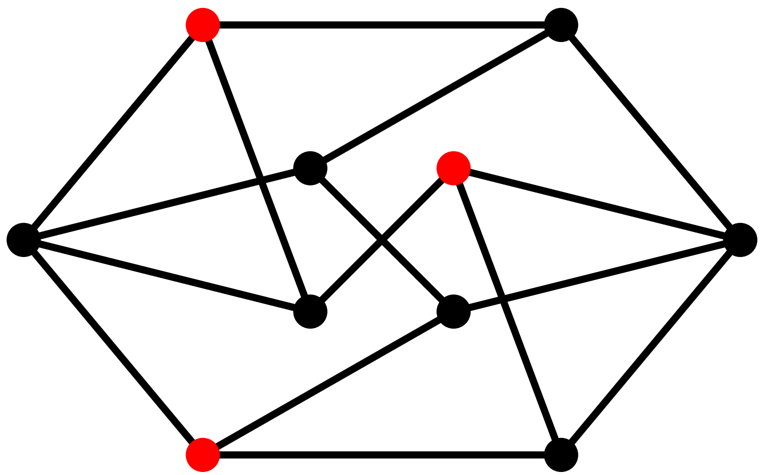
(4) # maximal independent sets of G

(5) # minimum maximal independent sets of G

A maximum independent set

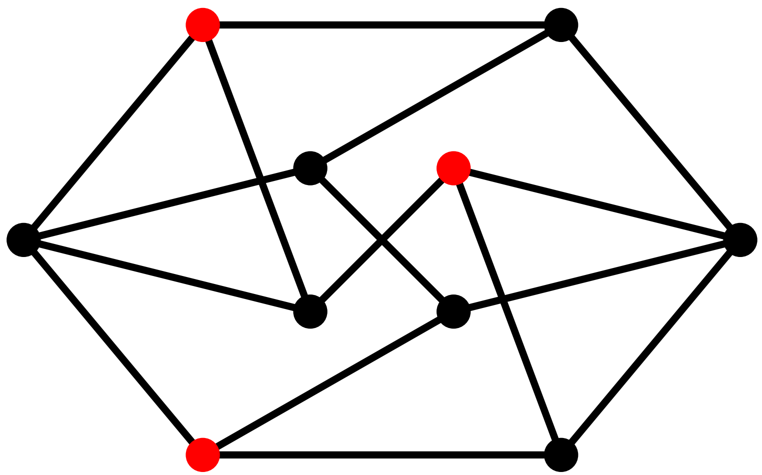
Setup $G = (V, E)$ a graph

Def. An independent set I of G is **maximum** if it has the largest size among all independent sets of G.



Setup $G = (V, E)$ a graph

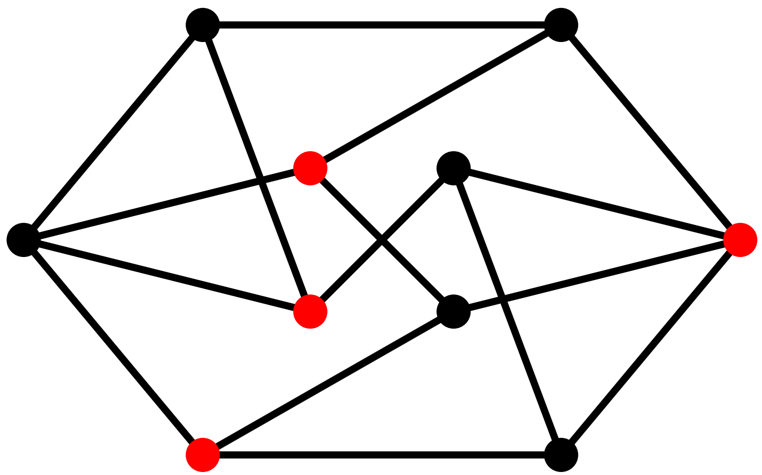
Def. An independent set I of G is **maximum** if it has the largest size among all independent sets of G.



Not maximum

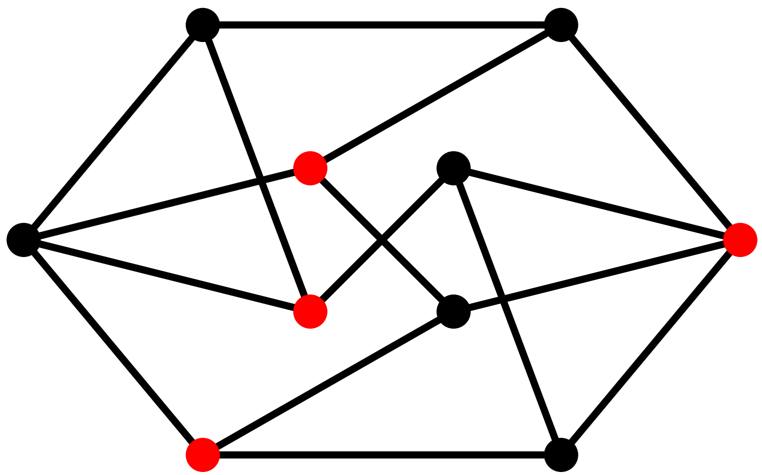
Setup $G = (V, E)$ a graph

Def. An independent set I of G is **maximum** if it has the largest size among all independent sets of G.



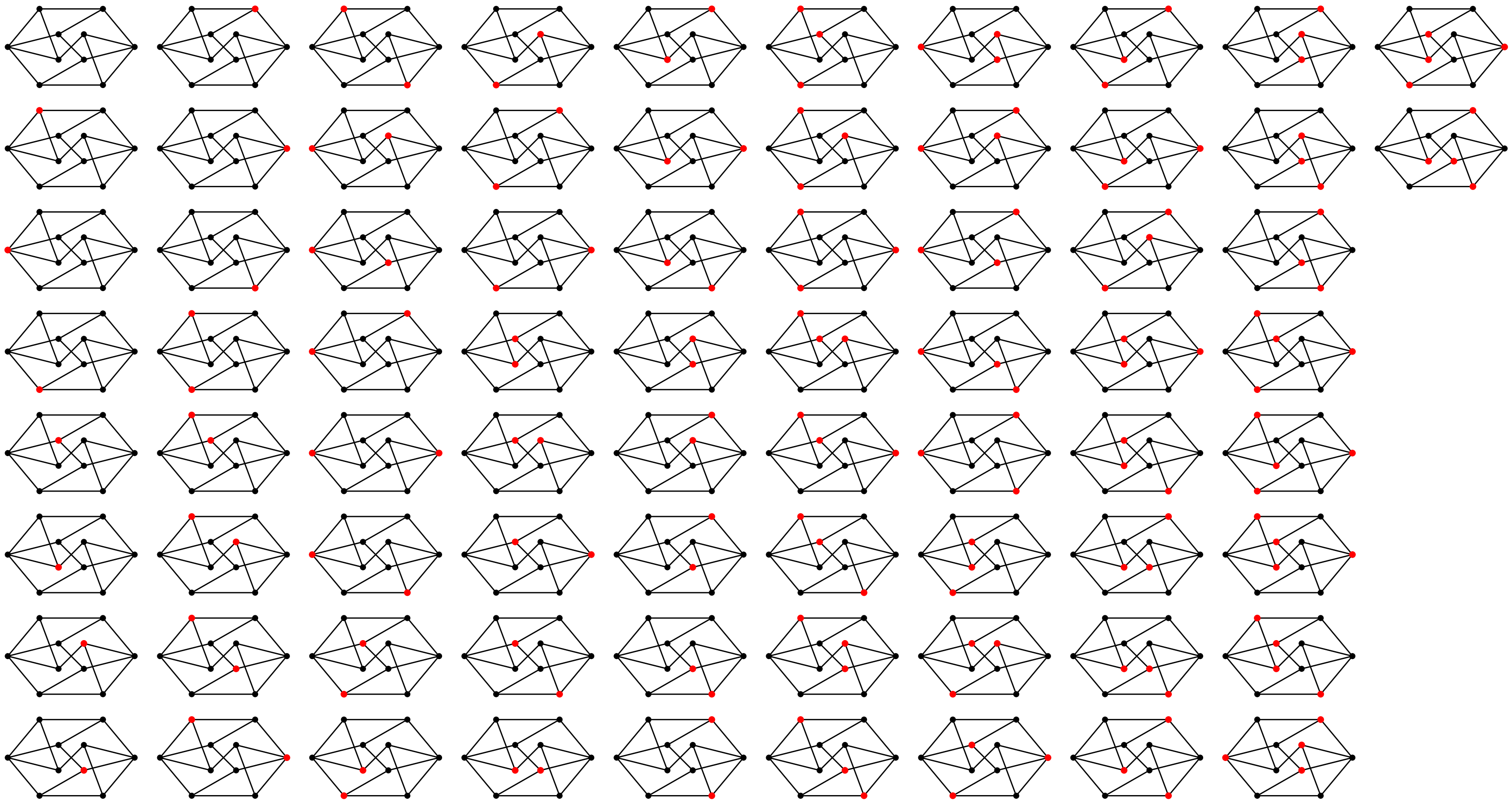
Setup $G = (V, E)$ a graph

Def. An independent set I of G is **maximum** if it has the largest size among all independent sets of G.



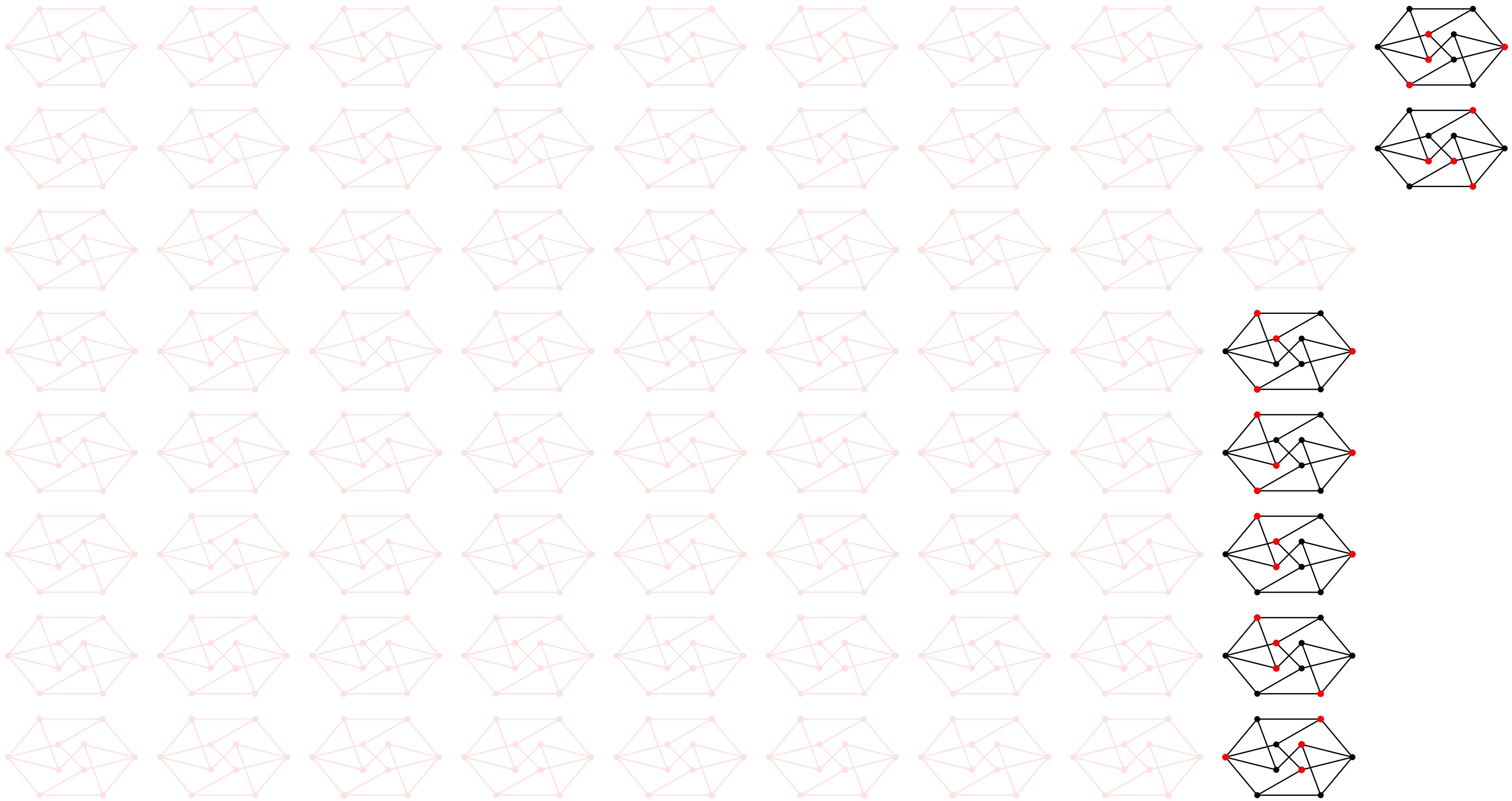
Maximum

Maximum independent sets



74 independent sets

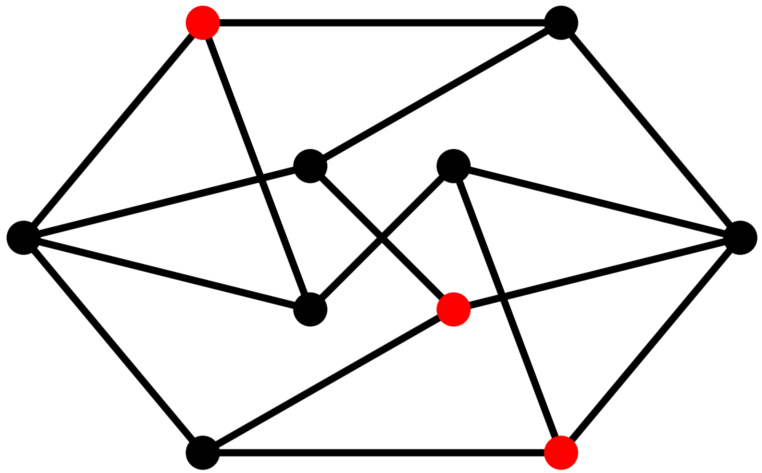
Maximum independent sets



74 independent sets 7 maximum independent sets

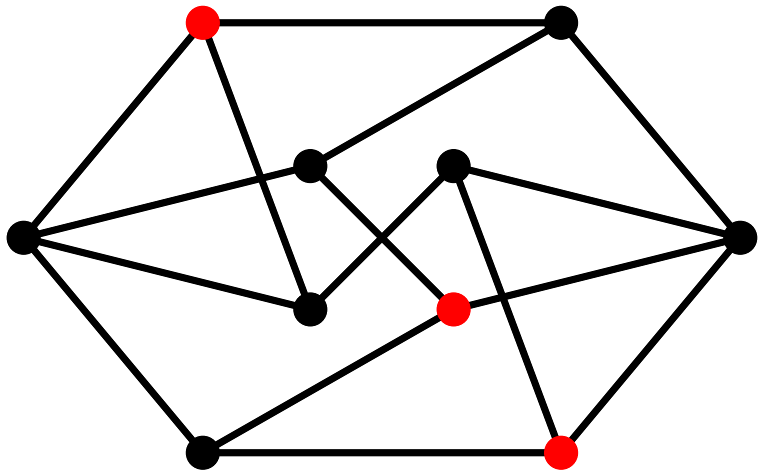
Setup $G = (V, E)$ a graph

Def. An independent set I of G is **maximal** if no proper superset of I is independent.



Setup $\quad G = (V, E)$ a graph

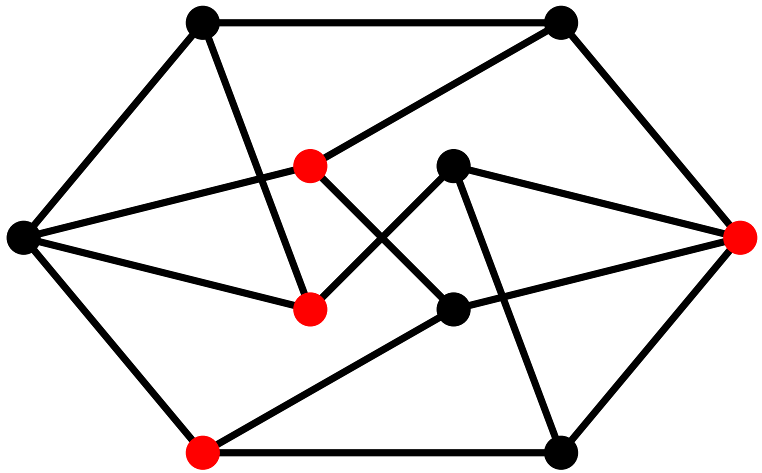
Def. $\quad$ An independent set I of G is **maximal** if no proper superset of I is independent.



Maximal

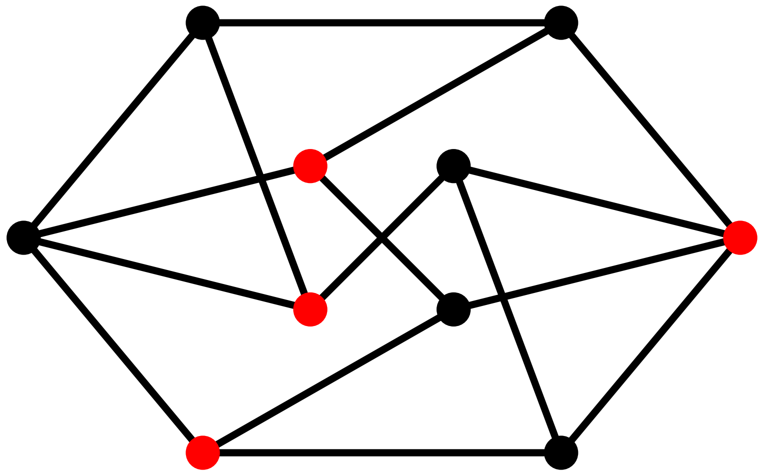
Setup $G = (V, E)$ a graph

Def. An independent set I of G is **maximal** if no proper superset of I is independent.



Setup $G = (V, E)$ a graph

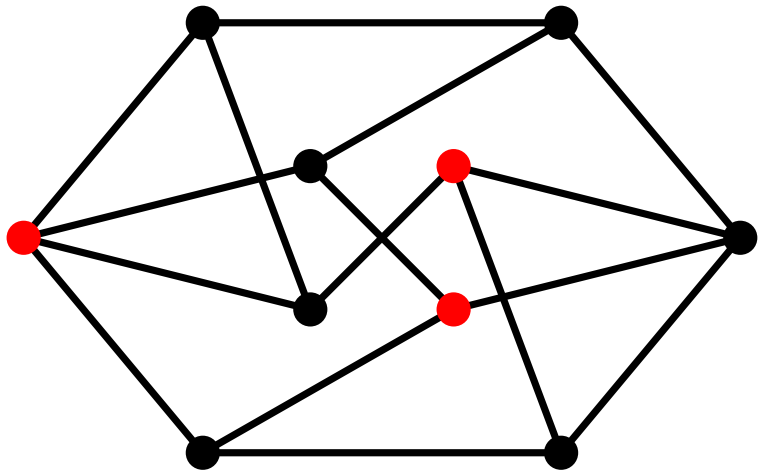
Def. An independent set I of G is **maximal** if no proper superset of I is independent.



Maximal

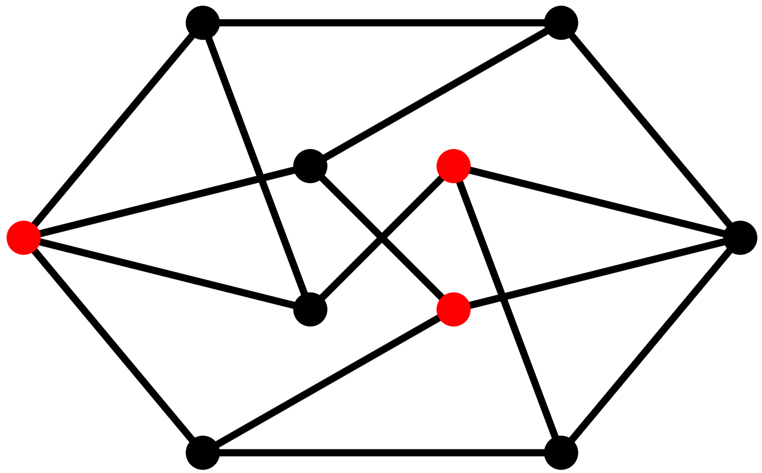
Setup $G = (V, E)$ a graph

Def. An independent set I of G is **maximal** if no proper superset of I is independent.



Setup $G = (V, E)$ a graph

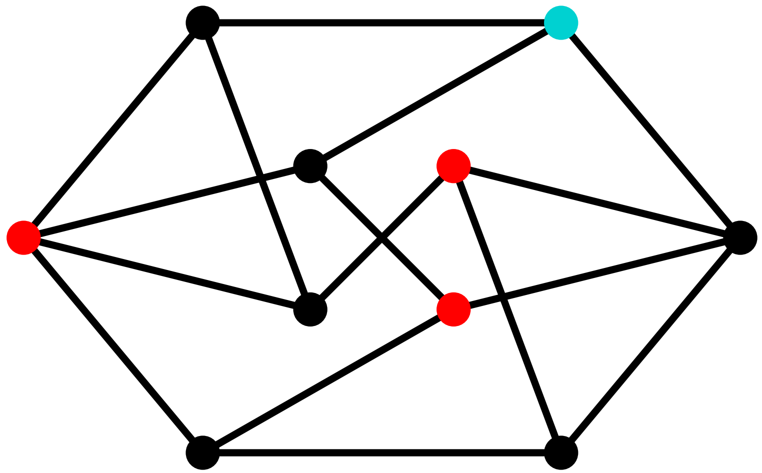
Def. An independent set I of G is **maximal** if no proper superset of I is independent.



Not maximal

Setup $G = (V, E)$ a graph

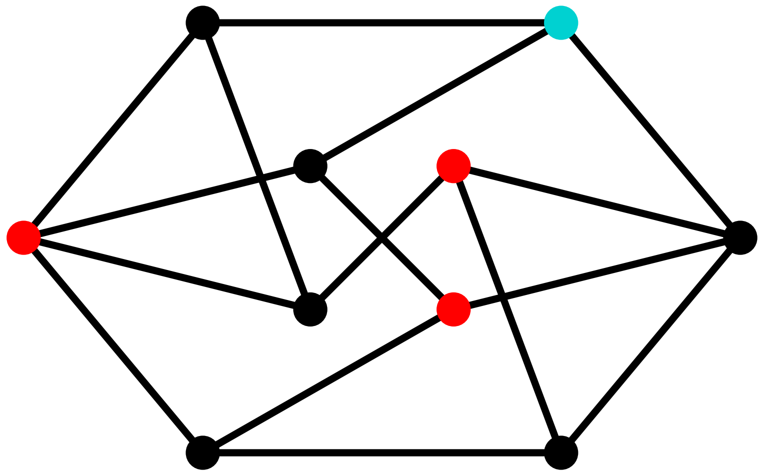
Def. An independent set I of G is **maximal** if no proper superset of I is independent.



Not maximal

Setup $G = (V, E)$ a graph

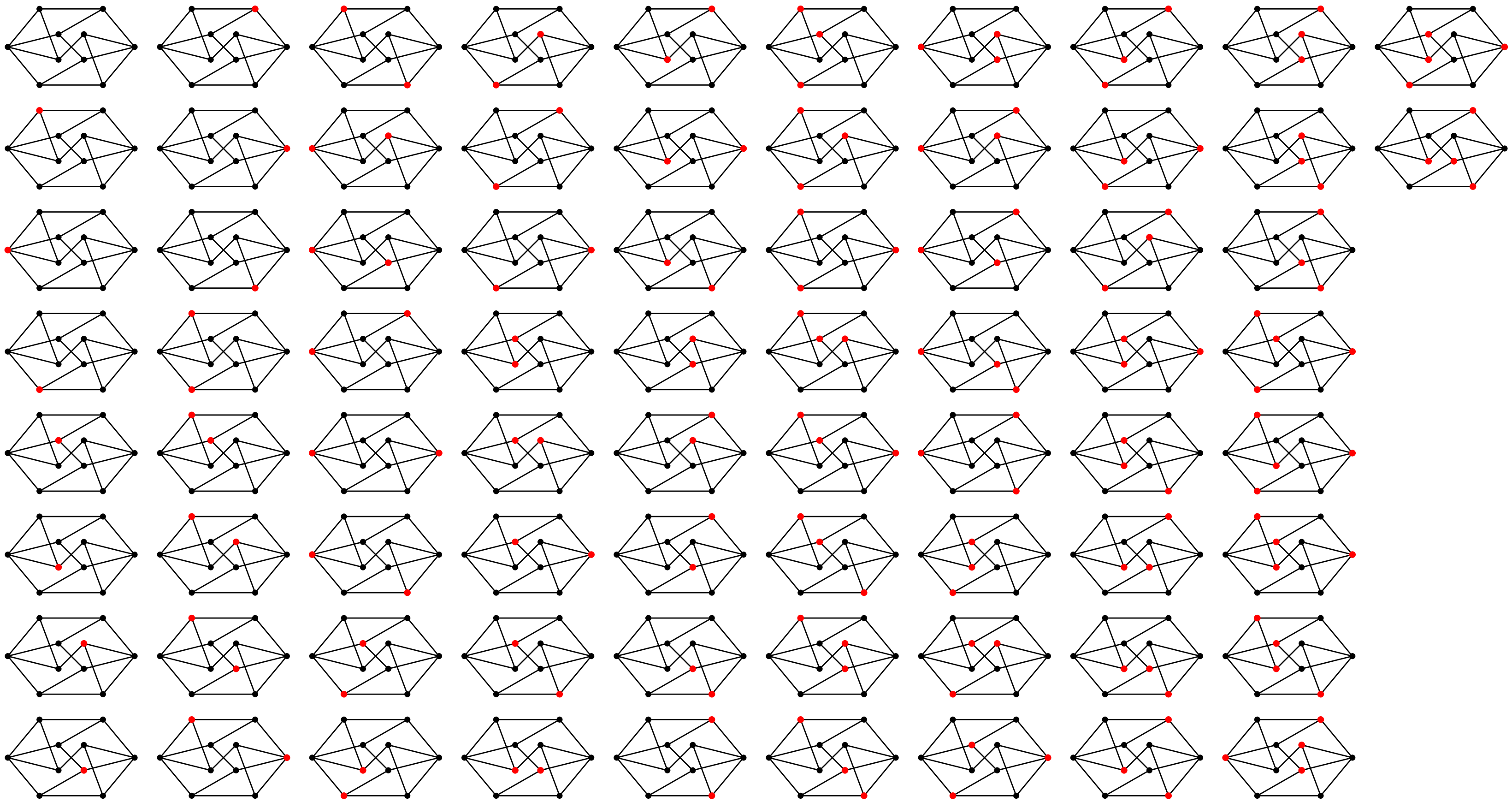
Def. An independent set I of G is **maximal** if no proper superset of I is independent.



Not maximal

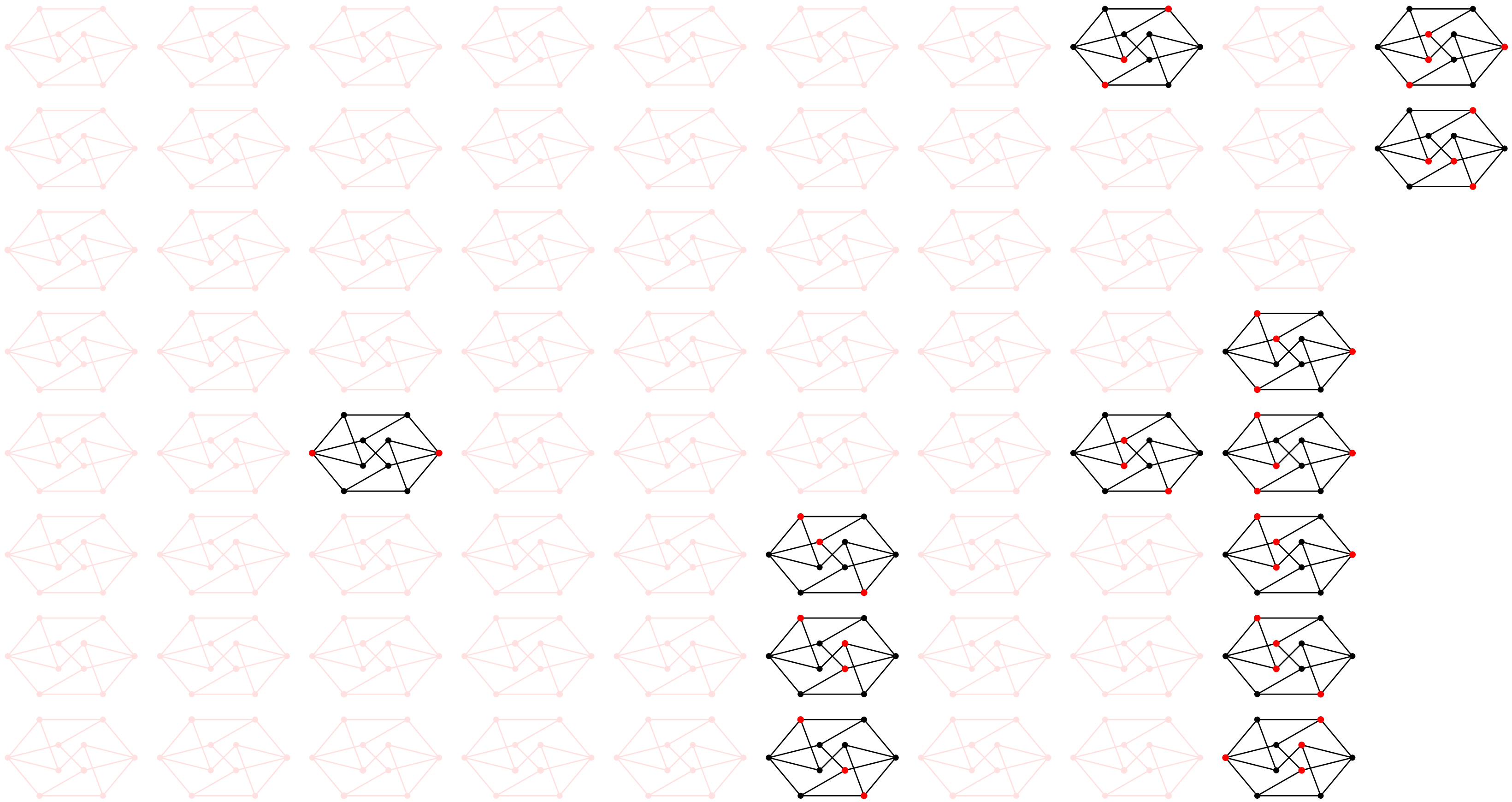
Also called an independent dominating set of G

Maximal independent sets



74 independent sets

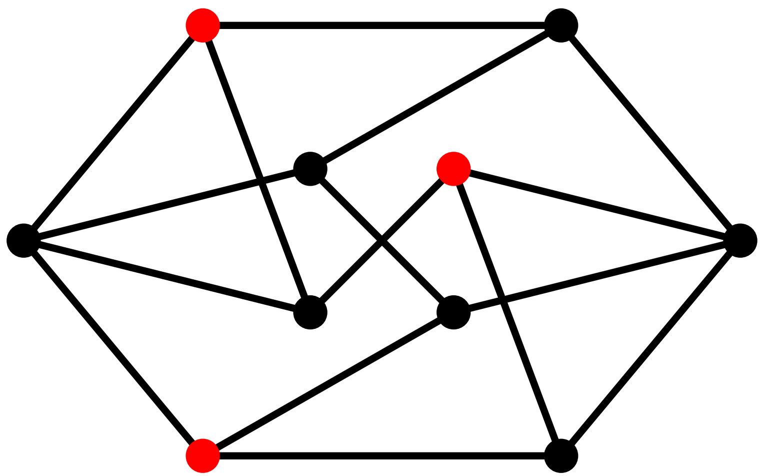
Maximal independent sets



74 independent sets 13 maximal independent sets

Setup $G = (V, E)$ a graph

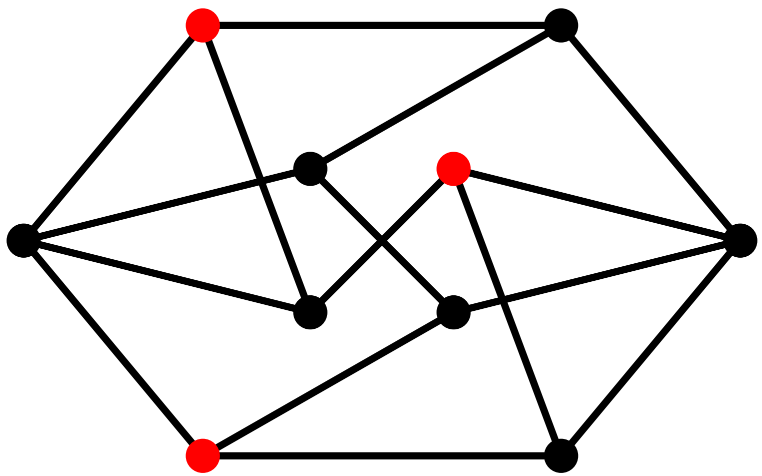
Def. An independent set I of G is **minimum maximal** if it is maximal and has the smallest size among all maximal independent sets of G.



maximal

Setup $G = (V, E)$ a graph

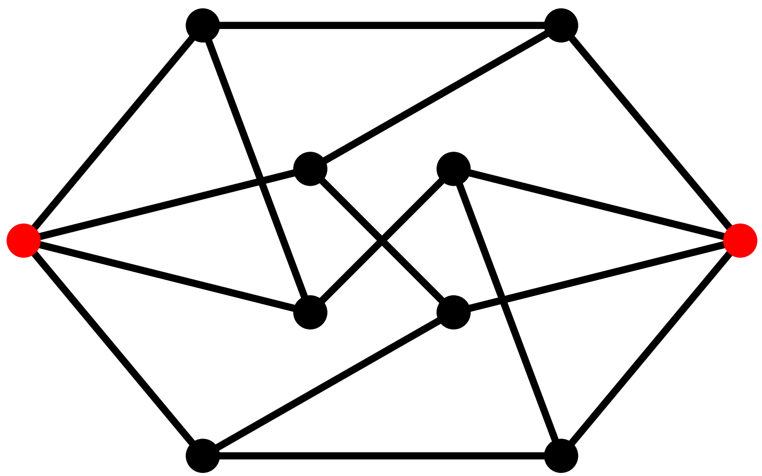
Def. An independent set I of G is **minimum maximal** if it is maximal and has the smallest size among all maximal independent sets of G.



Not minimum maximal

Setup $G = (V, E)$ a graph

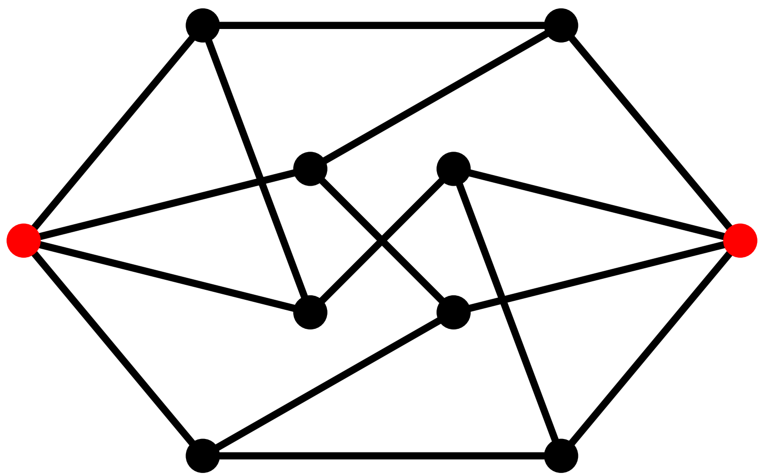
Def. An independent set I of G is **minimum maximal** if it is maximal and has the smallest size among all maximal independent sets of G.



A minimum maximal independent set

Setup $G = (V, E)$ a graph

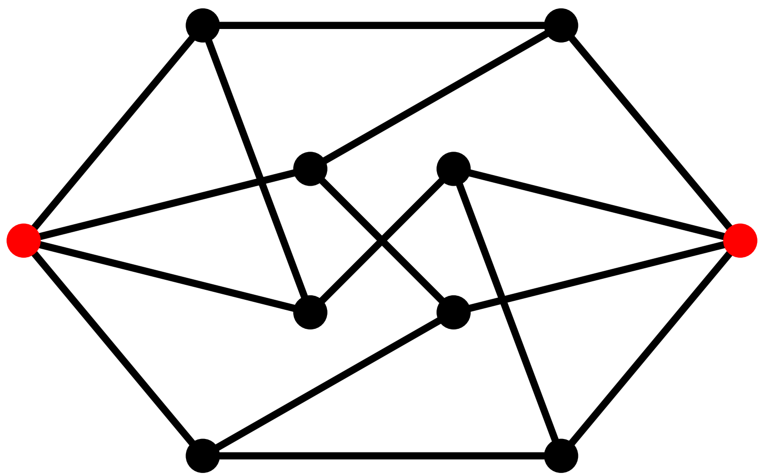
Def. An independent set I of G is **minimum maximal** if it is maximal and has the smallest size among all maximal independent sets of G.



Minimum maximal

Setup $G = (V, E)$ a graph

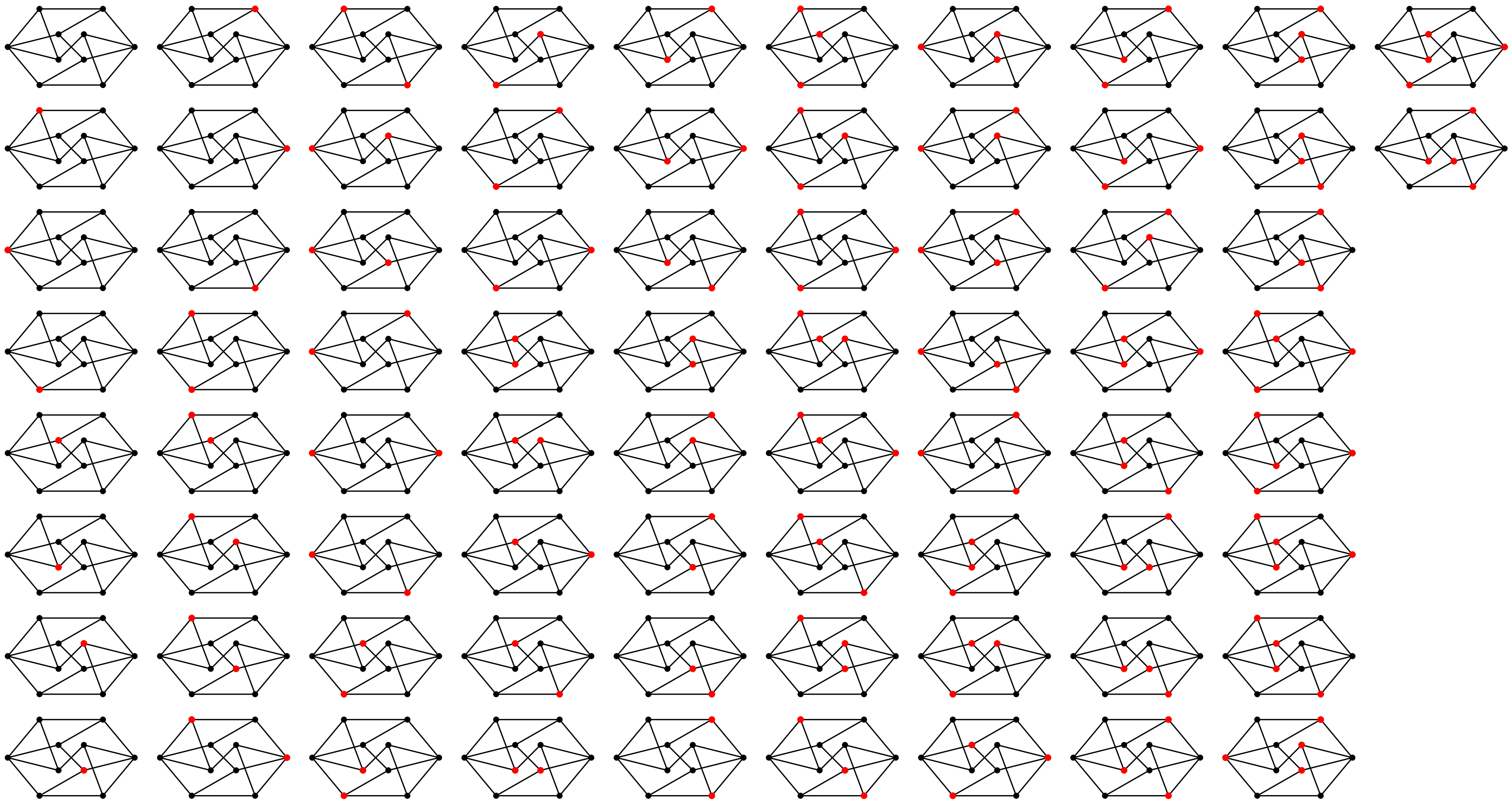
Def. An independent set I of G is **minimum maximal** if it is maximal and has the smallest size among all maximal independent sets of G.



Minimum maximal

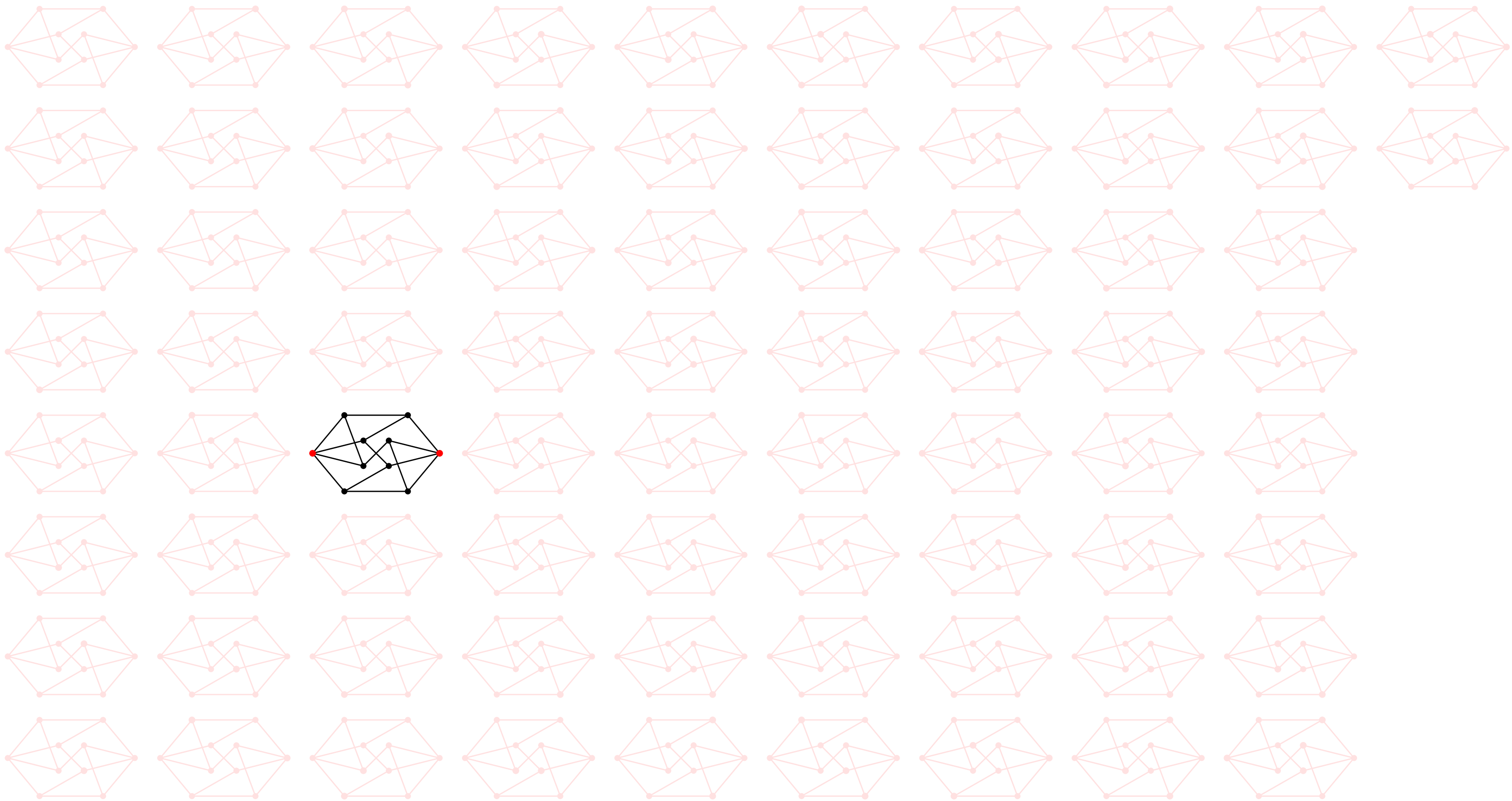
Also called a minimum independent dominating set of G

Minimum maximal independent sets



74 independent sets

Minimum maximal independent sets



74 independent sets

1 minimum maximal independent set

We study the following **counting problems**:

Input $G = (V, E)$ a graph

Output

(1) # independent sets of G **74**

(2) # maximum independent sets of G **7**

(3) # independent sets of G of fixed size

(4) # maximal independent sets of G **13**

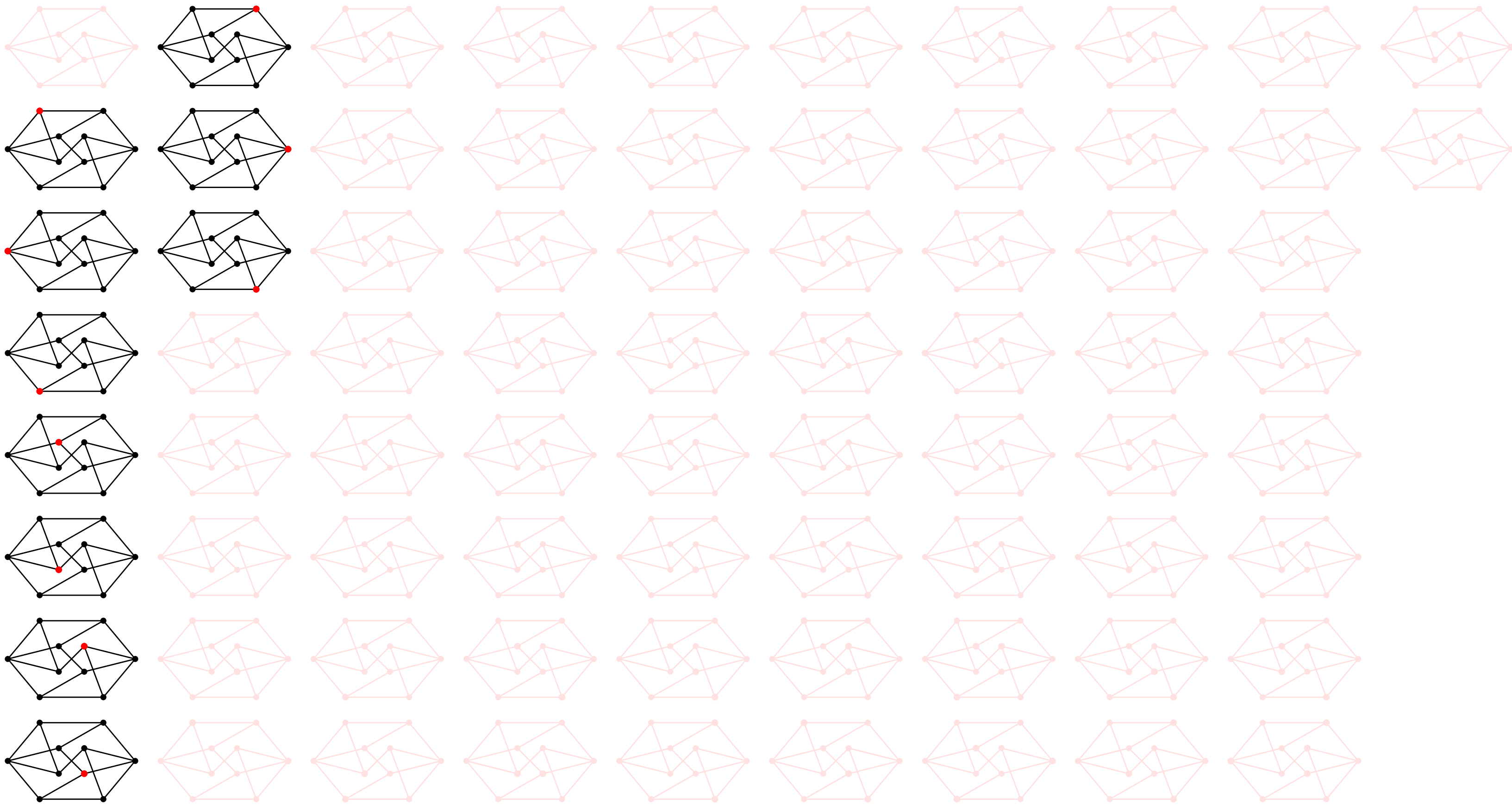
(5) # minimum maximal independent sets of G **1**

Independent sets of fixed size

74 independent sets

1 independent set of size 0

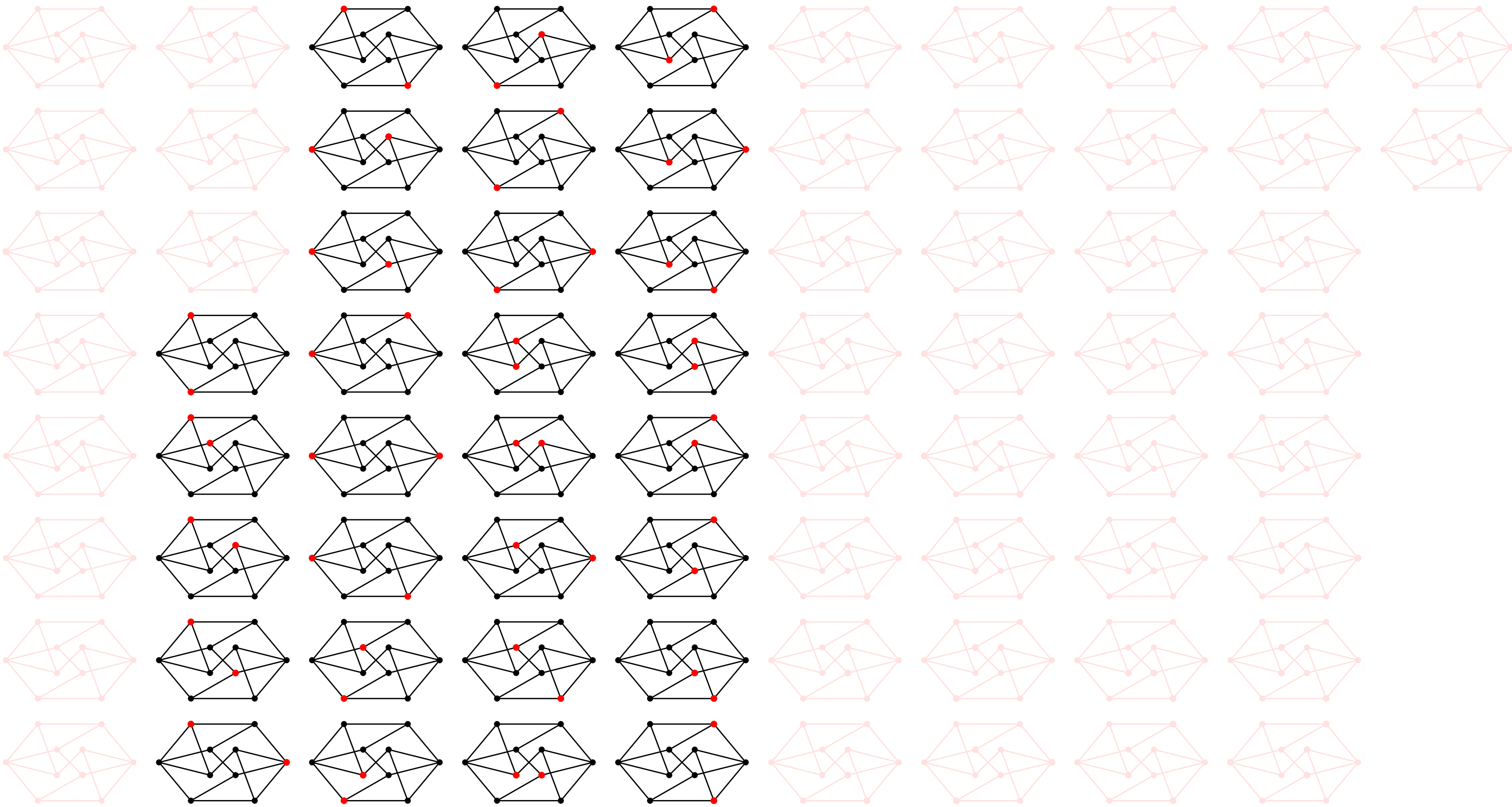
Independent sets of fixed size



74 independent sets

10 independent sets of size 1

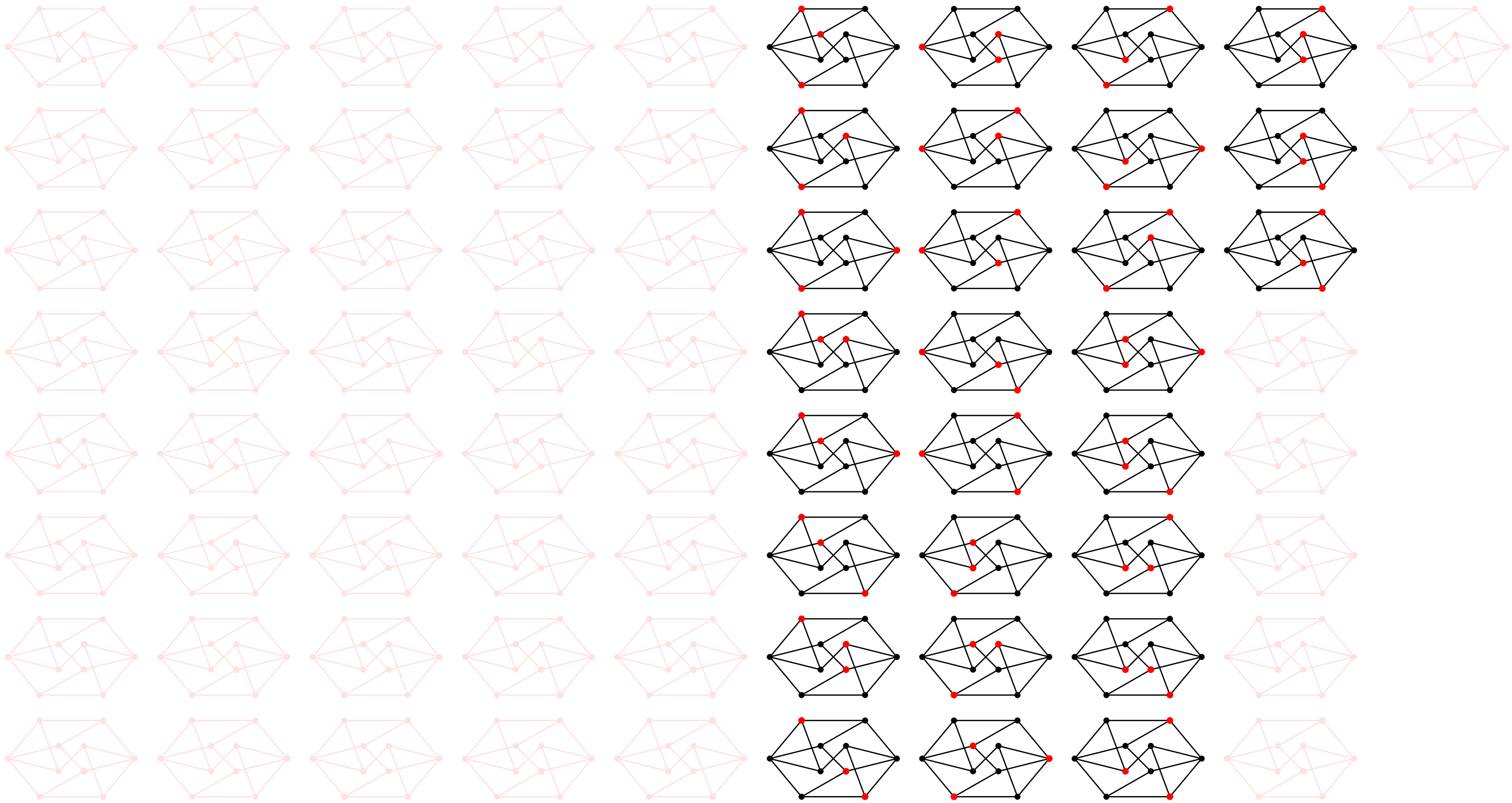
Independent sets of fixed size



74 independent sets

29 independent sets of size 2

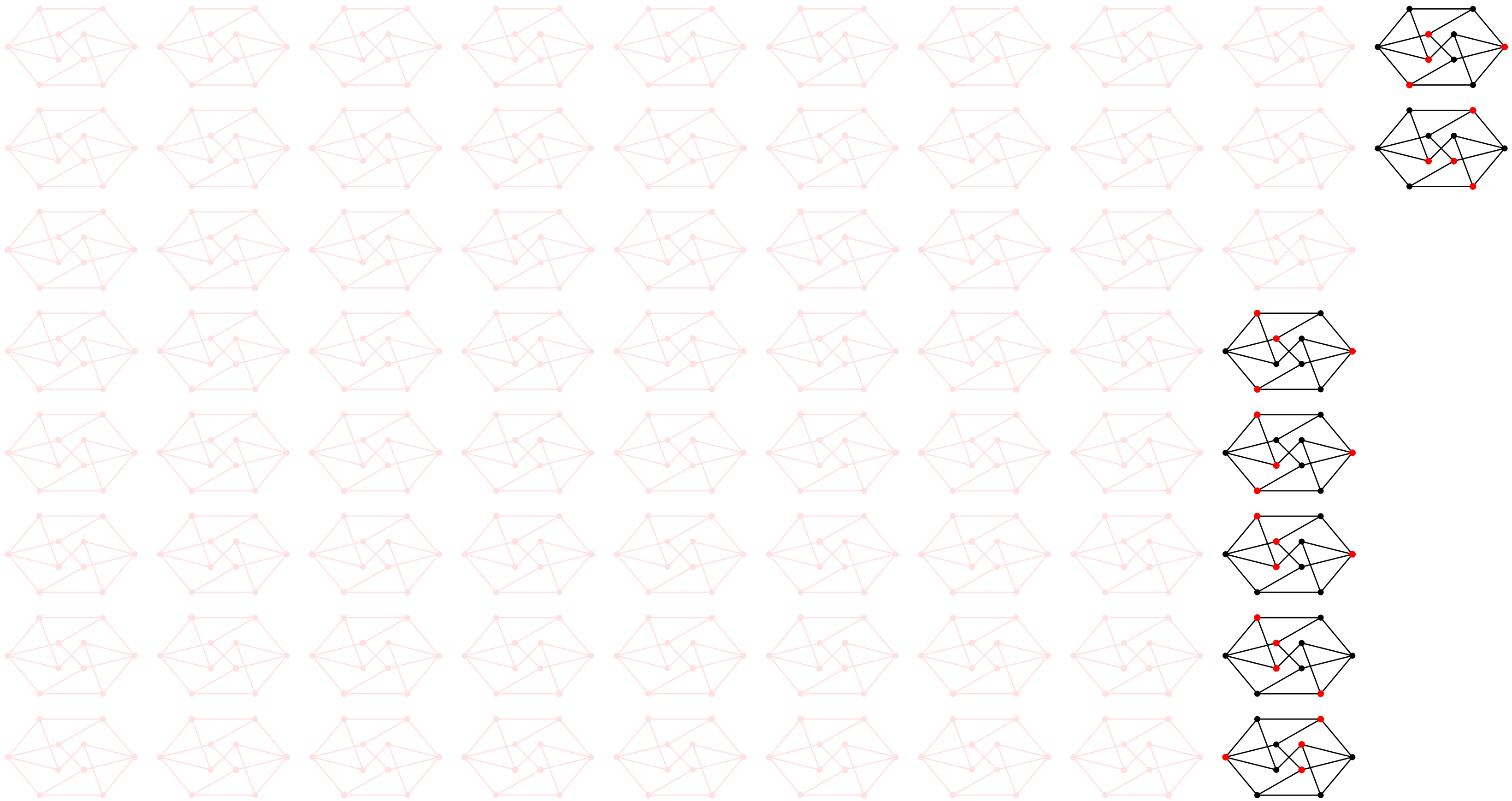
Independent sets of fixed size



74 independent sets

27 independent sets of size 3

Independent sets of fixed size



74 independent sets

7 independent sets of size 4

$\boxed{\text{Fact}}$ These counting problems are #P-complete (analoguous to NP-completeness).

$\Longrightarrow$ Cannot hope for a poly-time algorithms.

Fact These counting problems are #P-complete (analoguous to NP-completeness).

$\Longrightarrow$ Cannot hope for a poly-time algorithms even for

- the line graphs of bipartite graphs (Valiant '79)
- bipartite graphs (Provan & Ball '83)
- planar bipartite graphs of max deg 4 (Vadhan '01).

$\boxed{\text{Fact}}$ These counting problems are #P-complete (analoguous to NP-completeness).

$\Longrightarrow$ Cannot hope for a poly-time algorithms even for

- the line graphs of bipartite graphs (Valiant '79)
- bipartite graphs (Provan & Ball '83)
- planar bipartite graphs of max deg 4 (Vadhan '01).

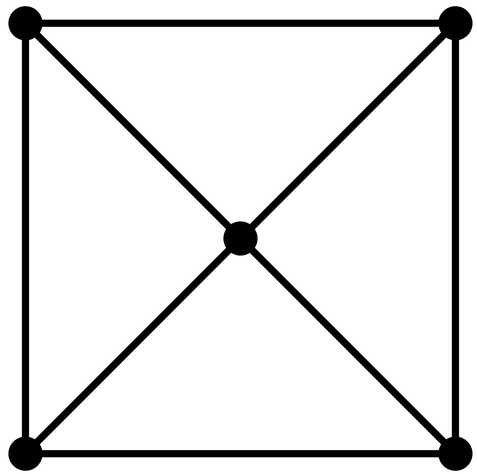
$\Longrightarrow$ Focus on another class of perfect graphs!!

Setup $G = (V, E)$ a graph

Def. G is **chordal**
if every induced cycle is of length three.

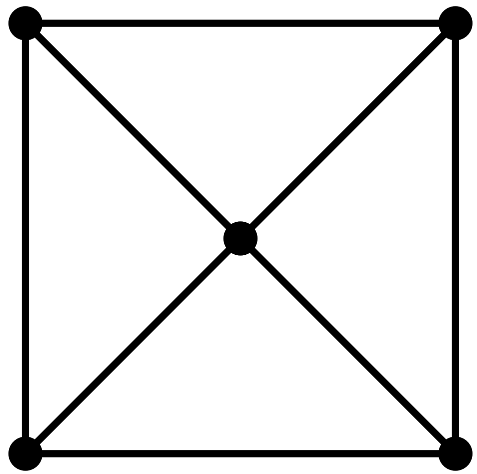
Setup $G = (V, E)$ a graph

Def. G is **chordal** if every induced cycle is of length three.



Setup $G = (V, E)$ a graph

Def. G is **chordal** if every induced cycle is of length three.

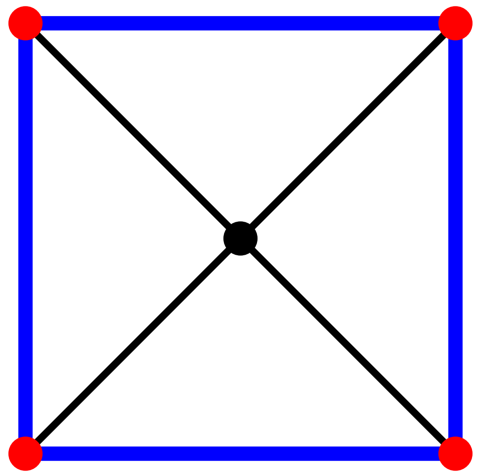


Not chordal

Setup $G = (V, E)$ a graph

Def. G is **chordal**

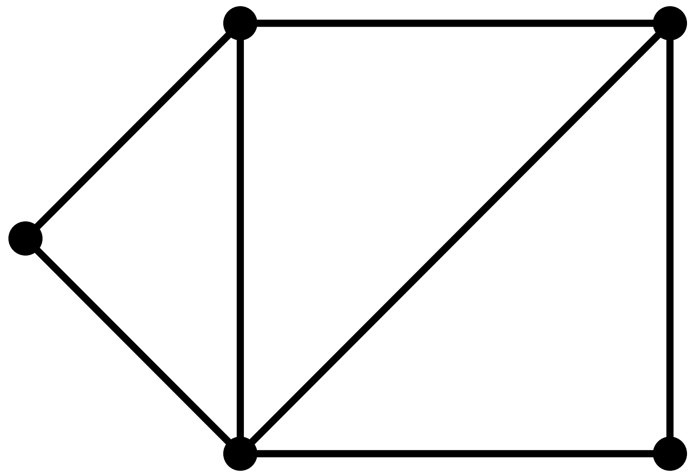
if every induced cycle is of length three.



Not chordal

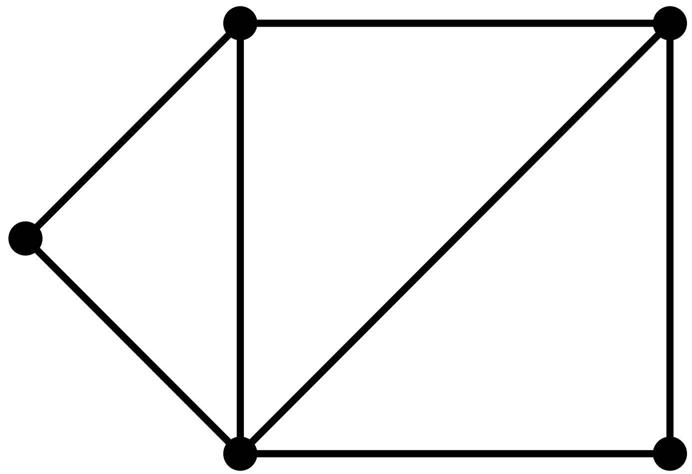
Setup $G = (V, E)$ a graph

Def. G is **chordal**
if every induced cycle is of length three.



Setup $G = (V, E)$ a graph

Def. G is **chordal** if every induced cycle is of length three.



Chordal

We study the following **counting problems**:

Input $G = (V, E)$ a chordal graph

Output (1) # independent sets of G

(2) # maximum independent sets of G

(3) # independent sets of G of fixed size

(4) # maximal independent sets of G

(5) # minimum maximal independent sets of G

We study the following **counting problems**:

Input $G = (V, E)$ a chordal graph

Output (1) # independent sets of G

(2) # maximum independent sets of G

(3) # independent sets of G of fixed size

(4) # maximal independent sets of G

(5) # minimum maximal independent sets of G

Rem "Finding one" is easy. (Gavril '72, Farber '82)

We obtain the following results for chordal graphs.

(1) # independent sets of G
(2) # maximum independent sets of G
(3) # independent sets of G of fixed size

(4) # maximal independent sets of G
(5) # minimum maximal independent sets of G

We obtain the following results for chordal graphs.

(1) # independent sets of G
(2) # maximum independent sets of G
(3) # independent sets of G of fixed size

$O(|V| + |E|)$ alg.

(4) # maximal independent sets of G
(5) # minimum maximal independent sets of G

#P-complete

We obtain the following results for chordal graphs.

(1) # independent sets of G
(2) # maximum independent sets of G
(3) # independent sets of G of fixed size

$O(|V| + |E|)$ alg.

(4) # maximal independent sets of G
(5) # minimum maximal independent sets of G

#P-complete

Given $G = (V, E)$ a (connected) chordal graph

Def A **clique tree** of G is a tree T st

(1) the nodes of $T =$ the maximal cliques of G,

(2) $\forall\, v \in V$,

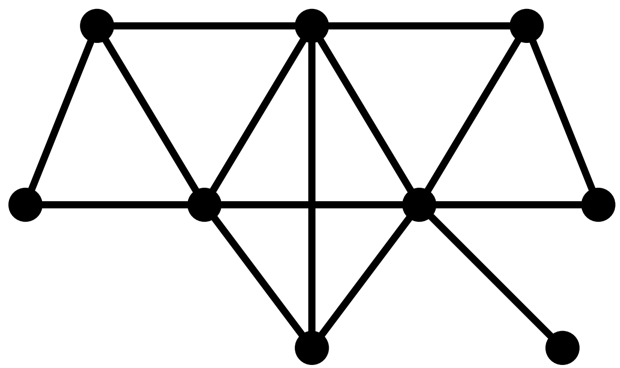
the nodes of T containing v induce a tree.

Given $G = (V, E)$ a (connected) chordal graph

Def A **clique tree** of G is a tree T st

(1) the nodes of T = the maximal cliques of G,

(2) $\forall\, v \in V$,
the nodes of T containing v induce a tree.



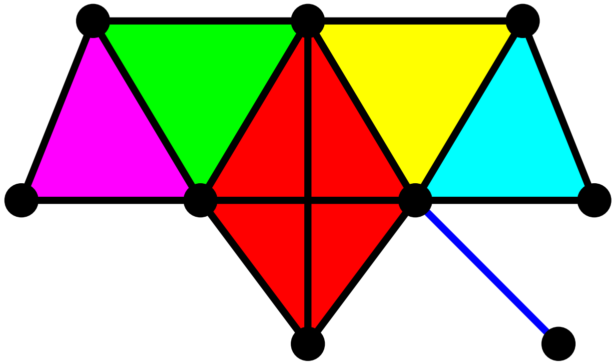
Given $G = (V, E)$ a (connected) chordal graph

Def A **clique tree** of G is a tree T st

(1) the nodes of T = the maximal cliques of G,

(2) $\forall\ v \in V$,

the nodes of T containing v induce a tree.



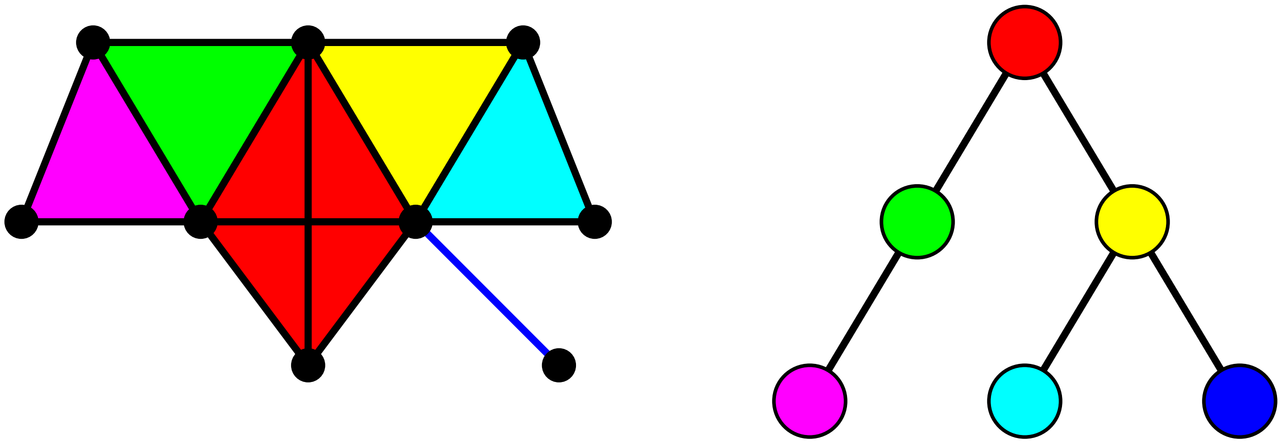
Given $G = (V, E)$ a (connected) chordal graph

Def A **clique tree** of G is a tree T st

(1) the nodes of T = the maximal cliques of G,

(2) $\forall v \in V$,

the nodes of T containing v induce a tree.



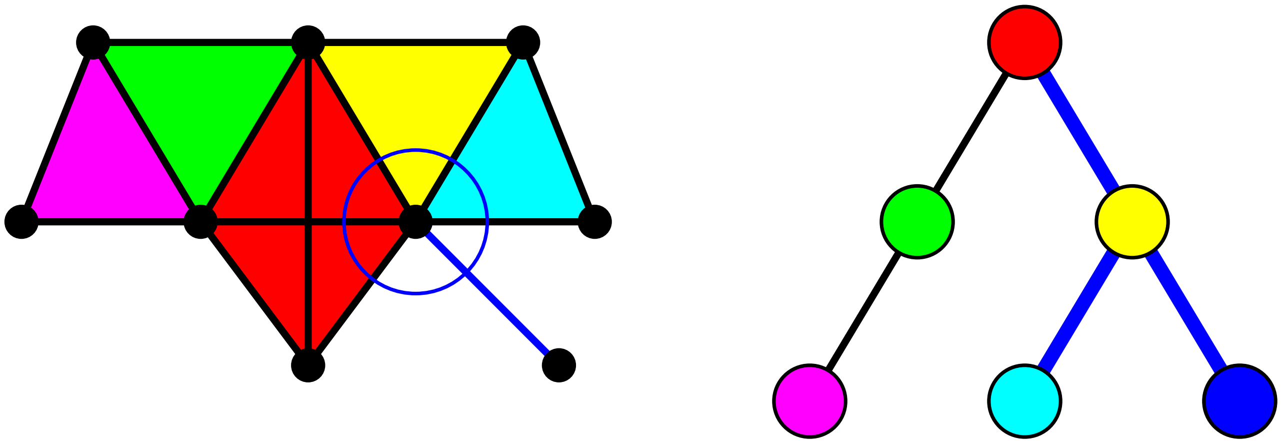
Given $G = (V, E)$ a (connected) chordal graph

Def A **clique tree** of G is a tree T st

(1) the nodes of T = the maximal cliques of G,

(2) $\forall\, v \in V$,

the nodes of T containing v induce a tree.



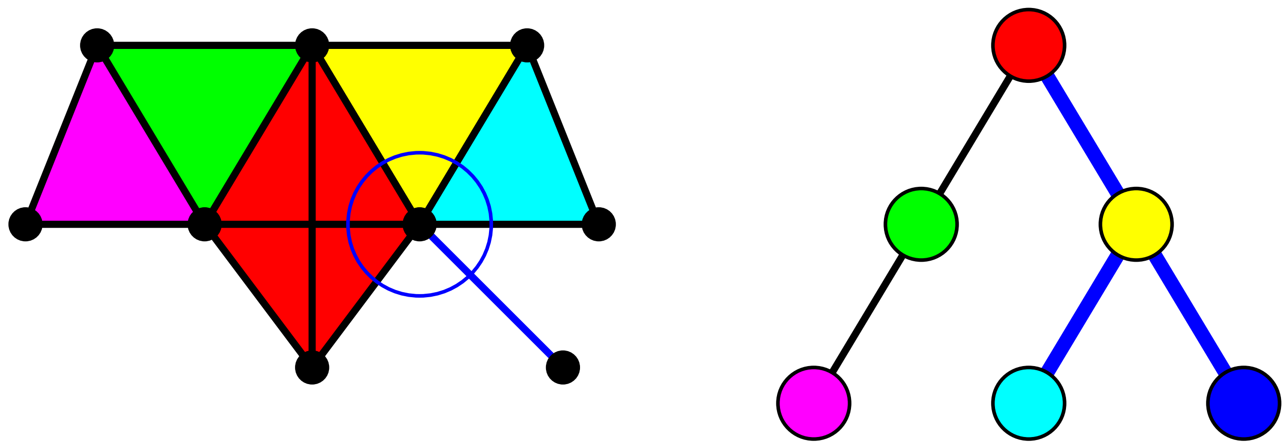
Given $G = (V, E)$ a (connected) chordal graph

Def A **clique tree** of G is a tree T st

(1) the nodes of T = the maximal cliques of G,

(2) $\forall\, v \in V$,

the nodes of T containing v induce a tree.

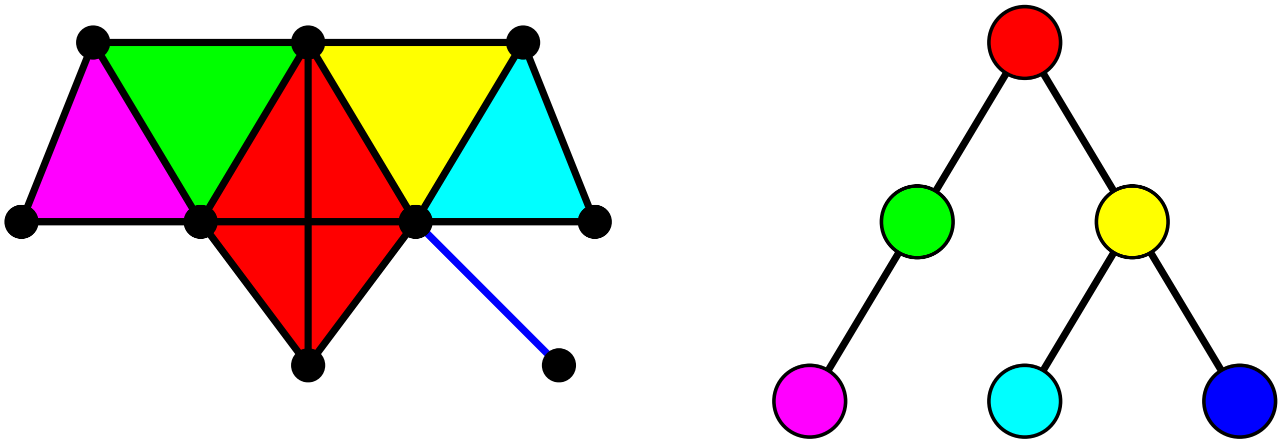


A clique tree can be computed in linear time.

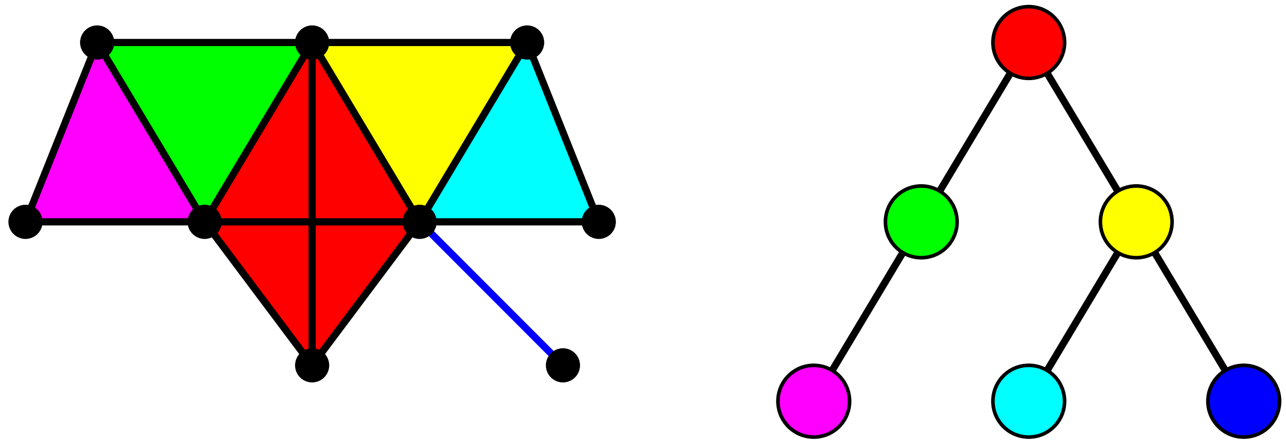
Given $\quad G = (V, E)$ a (connected) chordal graph,

Given $G = (V, E)$ a (connected) chordal graph,
T a clique tree of G, with root K

Given $G = (V, E)$ a (connected) chordal graph,
T a clique tree of G, with root K

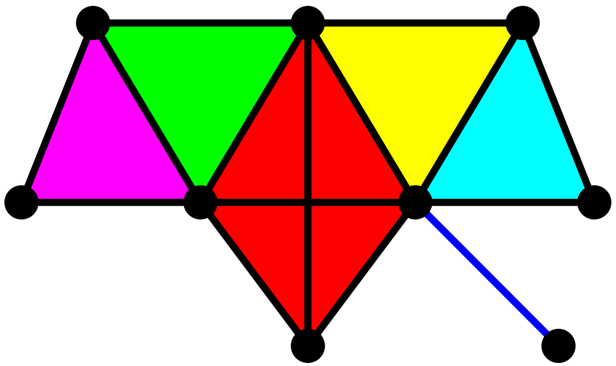


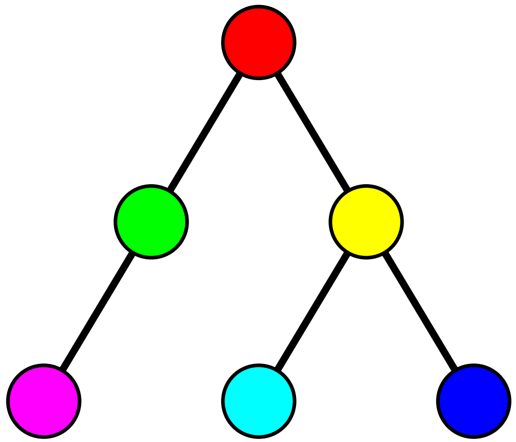
Given	$G = (V, E)$ a (connected) chordal graph, T a clique tree of G, with root K
Property	Every independent set of G either contains exactly one vertex in K or not.



Given $\quad G = (V, E)$ a (connected) chordal graph,
T a clique tree of G, with root K

Property $\quad$ Every independent set of G either
contains exactly one vertex in K or not.

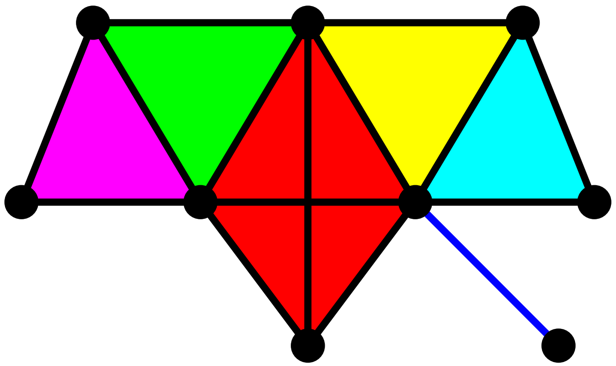


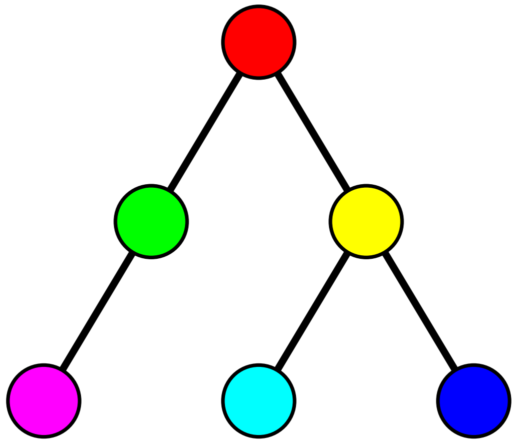


Leads to a recursive formula...

Given $G = (V, E)$ a (connected) chordal graph, T a clique tree of G, with root K

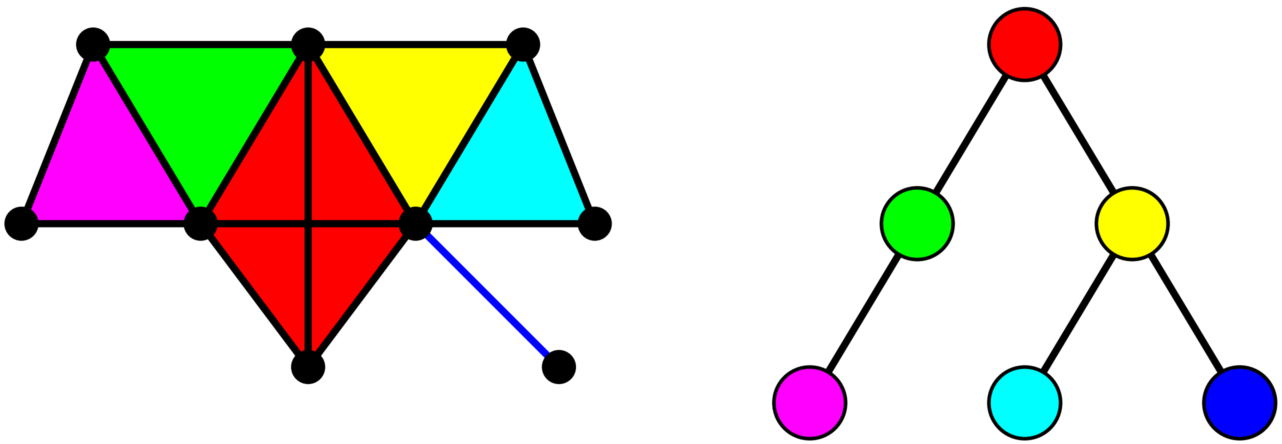
Lem # of independent sets of G =
# of independent sets of G excluding K +
$$\sum_{v \in K} (\text{\# of indepedent sets of } G \text{ containing } v)$$





Given $G = (V, E)$ a (connected) chordal graph,
T a clique tree of G with root K
K' a node of T

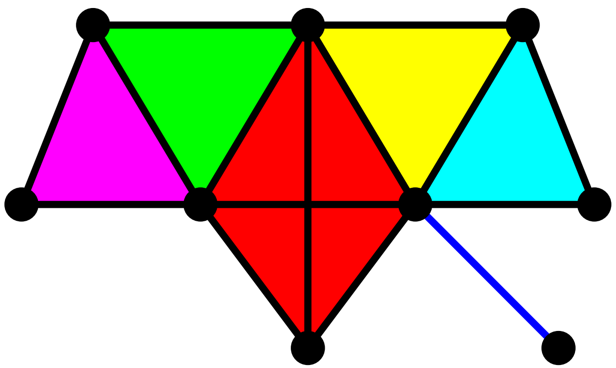
Notation $T(K')$ the subtree of G rooted at K'

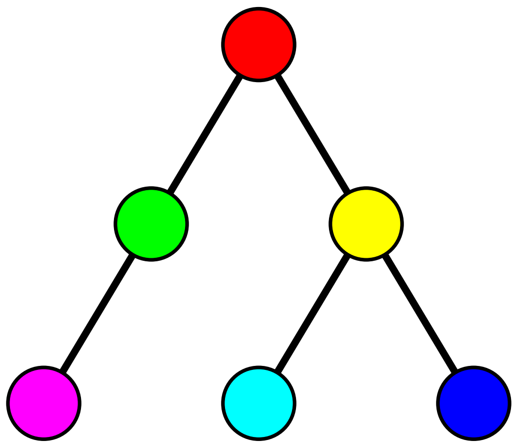


Given $G = (V, E)$ a (connected) chordal graph,
T a clique tree of G with root K
K' a node of T

Notation $T(K')$ the subtree of G rooted at K'

Property $T(K')$ is a clique tree of some chordal graph.

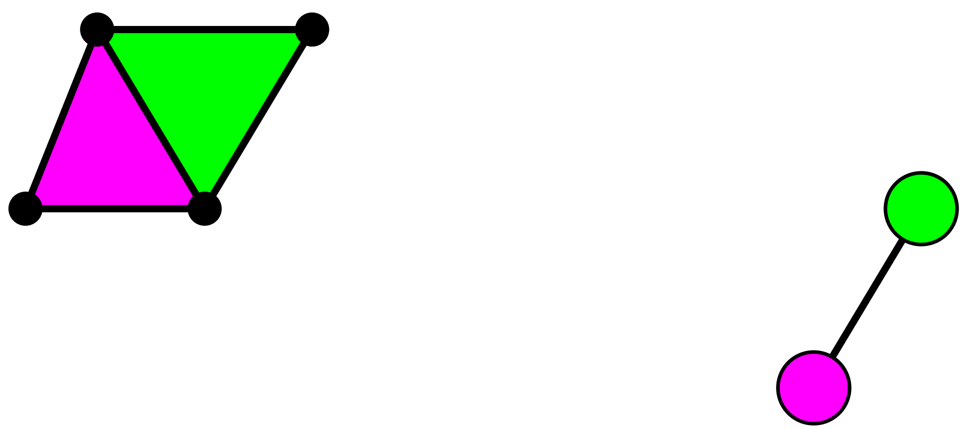




Given $G = (V, E)$ a (connected) chordal graph,
T a clique tree of G with root K
K' a node of T

Notation $T(K')$ the subtree of G rooted at K'

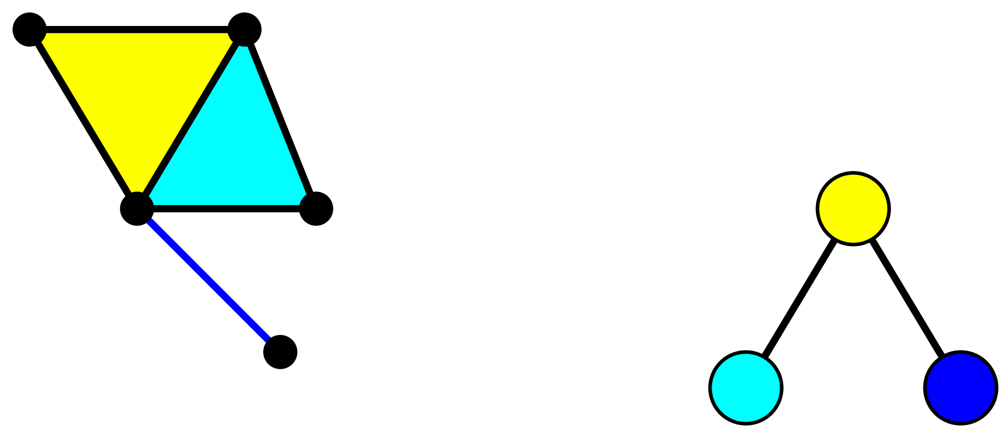
Property $T(K')$ is a clique tree of some chordal graph.



Given $G = (V, E)$ a (connected) chordal graph,
T a clique tree of G with root K
K' a node of T

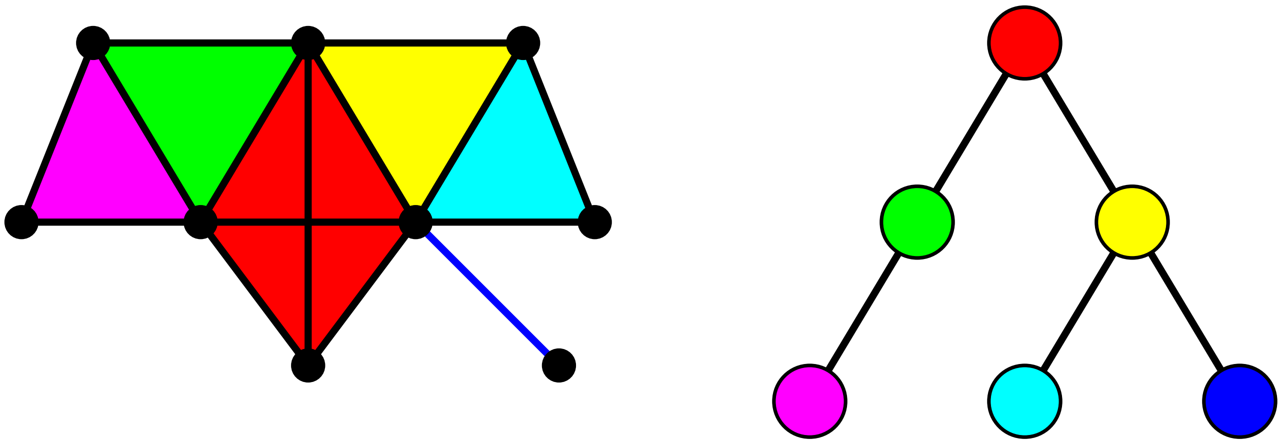
Notation $T(K')$ the subtree of G rooted at K'

Property $T(K')$ is a clique tree of some chordal graph.



Given $G = (V, E)$ a (connected) chordal graph,
T a clique tree of G, with root K,
$K_1, \ldots, K_r$ the children of K in T

Lem # of independent sets of G excluding K =
$$\prod_i (\text{\# of independent sets of } G \text{ excluding } K \cap K_i)$$



Let G be a chordal graph and T be a rooted clique tree of G. For a maximal clique K of G which is not a leaf of the clique tree, let $K_1, \ldots, K_\ell$ be the children of K in T. Furthermore, let $v \in K$. Then, the following identities hold.

$$\mathcal{IS}(G(K)) = \overline{\mathcal{IS}}(G(K), K) \,\dot\cup \dot{\bigcup_{v \in K}} \mathcal{IS}(G(K), v);$$

$$\mathcal{IS}(G(K), v) = \{S \cup \{v\} \mid S = \bigcup_{i=1}^{\ell} S_i, S_i \in \left\{ \begin{array}{ll} \mathcal{IS}(G(K_i), v) & \text{if } v \in K_i \\ \overline{\mathcal{IS}}(G(K_i), K \cap K_i) & \text{otherwise} \end{array} \right\}\};$$

$$\overline{\mathcal{IS}}(G(K), K) = \{S \mid S = \bigcup_{i=1}^{\ell} S_i, S_i \in \overline{\mathcal{IS}}(G(K_i), K \cap K_i)\};$$

$$\overline{\mathcal{IS}}(G(K_i), K \cap K_i) = \overline{\mathcal{IS}}(G(K_i), K_i) \,\dot\cup \dot{\bigcup_{u \in K_i \setminus K}} \mathcal{IS}(G(K_i), u) \quad \text{for each } i \in \{1, \ldots, \ell\},$$

where $\mathcal{IS}(G)$ denotes the family of independent sets in G, $\mathcal{IS}(G, v)$ denotes for a vertex v the family of independent sets in G including v, $\overline{\mathcal{IS}}(G, U)$ denotes the family of independent sets in G including no vertex of U for a vertex set U.

Recursive formula

Let G be a chordal graph and T be a rooted clique tree of G. For a maximal clique K of G which is not a leaf of the clique tree, let $K_1, \dots, K_\ell$ be the children of K in T. Furthermore, let $v \in K$. Then, the following identities hold.

$$\mathcal{IS}(G(K)) = \overline{\mathcal{IS}}(G(K), K) \,\dot\cup \dot{\bigcup_{v \in K}} \mathcal{IS}(G(K), v);$$

$$\mathcal{IS}(G(K), v) = \{S \cup \{v\} \mid S = \bigcup_{i=1}^{\ell} S_i, S_i \in \left\{ \begin{array}{ll} \mathcal{IS}(G(K_i), v) & \text{if } v \in K_i \\ \overline{\mathcal{IS}}(G(K_i), K \cap K_i) & \text{otherwise} \end{array} \right\}\};$$

$$\overline{\mathcal{IS}}(G(K), K) = \{S \mid S = \bigcup_{i=1}^{\ell} S_i, S_i \in \overline{\mathcal{IS}}(G(K_i), K \cap K_i)\};$$

$$\overline{\mathcal{IS}}(G(K_i), K \cap K_i) = \overline{\mathcal{IS}}(G(K_i), K_i) \,\dot\cup \dot{\bigcup_{u \in K_i \setminus K}} \mathcal{IS}(G(K_i), u) \quad \text{for each } i \in \{1, \dots, \ell\},$$

where $\mathcal{IS}(G)$ denotes the family of independent sets in G, $\mathcal{IS}(G, v)$ denotes for a vertex v the family of independent sets in G including v, $\overline{\mathcal{IS}}(G, U)$ denotes the family of independent sets in G including no vertex of U for a vertex set U.

$$\Downarrow$$

A detailed analysis yields a linear-time algorithm to count the independent sets in a chordal graph!

Recursive formula

Let G be a chordal graph and T be a rooted clique tree of G. For a maximal clique K of G which is not a leaf of the clique tree, let $K_1, \ldots, K_\ell$ be the children of K in T. Furthermore, let $v \in K$. Then, the following identities hold.

$$\mathcal{IS}(G(K)) = \overline{\mathcal{IS}}(G(K), K) \,\dot\cup \dot{\bigcup_{v \in K}} \mathcal{IS}(G(K), v);$$

$$\mathcal{IS}(G(K), v) = \{S \cup \{v\} \mid S = \bigcup_{i=1}^{\ell} S_i, S_i \in \begin{cases} \mathcal{IS}(G(K_i), v) & \text{if } v \in K_i \\ \overline{\mathcal{IS}}(G(K_i), K \cap K_i) & \text{otherwise} \end{cases}\};$$

$$\overline{\mathcal{IS}}(G(K), K) = \{S \mid S = \bigcup_{i=1}^{\ell} S_i, S_i \in \overline{\mathcal{IS}}(G(K_i), K \cap K_i)\};$$

$$\overline{\mathcal{IS}}(G(K_i), K \cap K_i) = \overline{\mathcal{IS}}(G(K_i), K_i) \,\dot\cup \dot{\bigcup_{u \in K_i \setminus K}} \mathcal{IS}(G(K_i), u) \quad \text{for each } i \in \{1, \ldots, \ell\},$$

where $\mathcal{IS}(G)$ denotes the family of independent sets in G, $\mathcal{IS}(G, v)$ denotes for a vertex v the family of independent sets in G including v, $\overline{\mathcal{IS}}(G, U)$ denotes the family of independent sets in G including no vertex of U for a vertex set U.

$$\Downarrow$$

A detailed analysis yields a linear-time algorithm to count the independent sets in a chordal graph!

Thm The independent sets in a chordal graph $G = (V, E)$ can be counted in $O(|V| + |E|)$ time.

We obtain the following results for chordal graphs.

(1) # independent sets of G
(2) # maximum independent sets of G
(3) # independent sets of G of fixed size

$O(|V| + |E|)$ alg.

(4) # maximal independent sets of G
(5) # minimum maximal independent sets of G

#P-complete

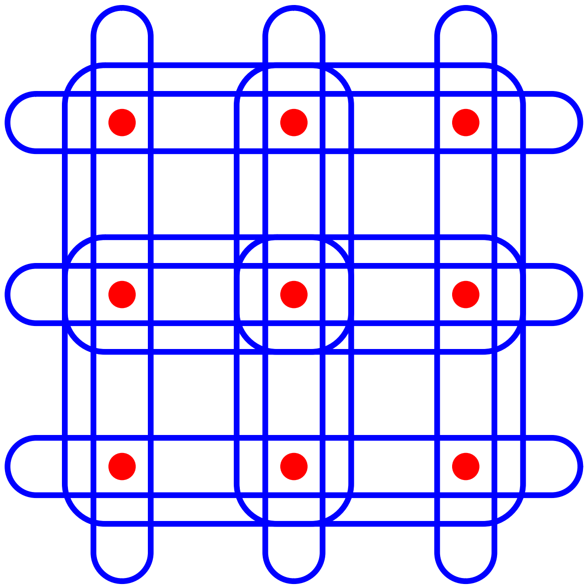
We use the following counting problem
(a counting version of the set cover problem).

Given	X a finite set, $\mathcal{S} \subseteq 2^{X}$ a family
Output	The number of subfamilies $\mathcal{S}' \subseteq \mathcal{S}$ st $\bigcup_{Y \in \mathcal{S}'} Y = X$.

We use the following counting problem
(a counting version of the set cover problem).

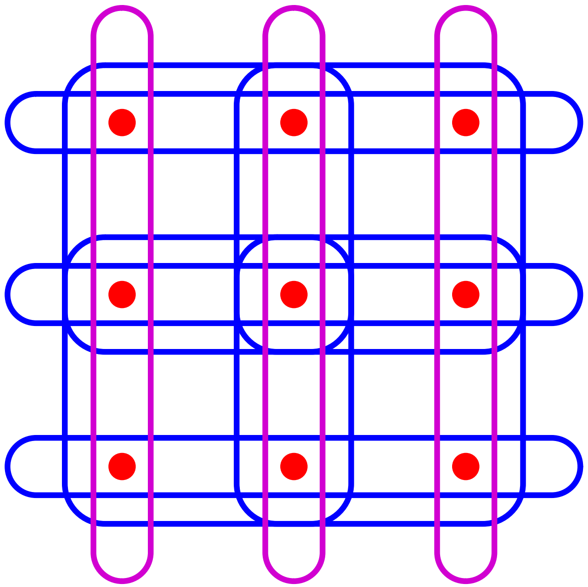
Given X a finite set, $\mathcal{S} \subseteq 2^X$ a family

Output The number of subfamilies $\mathcal{S}' \subseteq \mathcal{S}$ st $\bigcup_{Y \in \mathcal{S}'} Y = X$.



We use the following counting problem
(a counting version of the set cover problem).

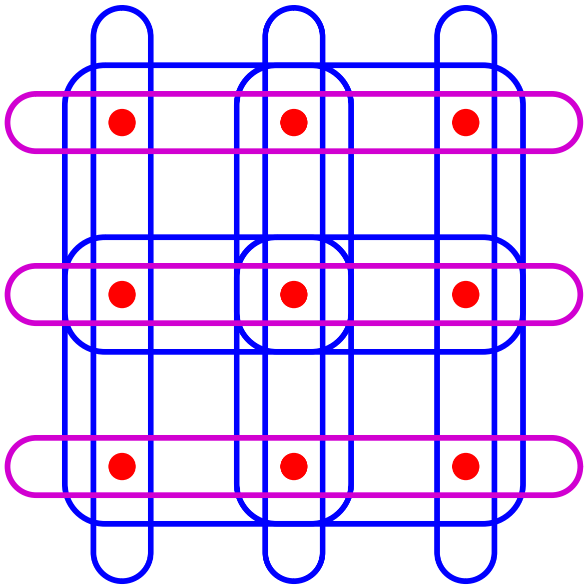
Given	X a finite set, $\mathcal{S} \subseteq 2^{\mathrm{X}}$ a family
Output	The number of subfamilies $\mathcal{S}' \subseteq \mathcal{S}$ st $\bigcup_{\mathrm{Y} \in \mathcal{S}'} \mathrm{Y} = \mathrm{X}$.



We use the following counting problem
(a counting version of the set cover problem).

Given X a finite set, $\mathcal{S} \subseteq 2^X$ a family

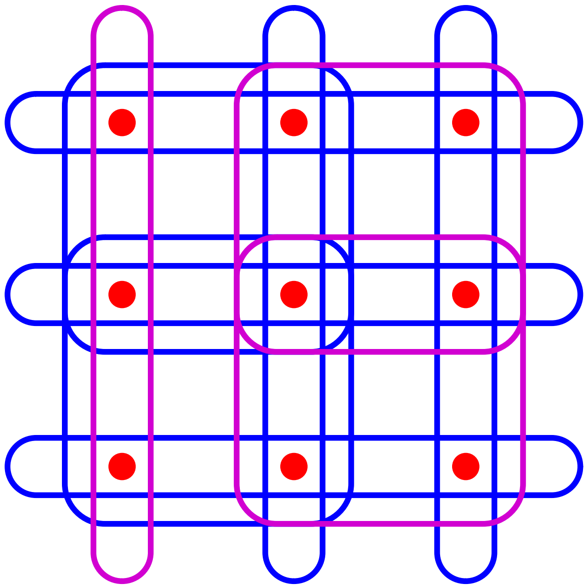
Output The number of subfamilies $\mathcal{S}' \subseteq \mathcal{S}$ st $\bigcup_{Y \in \mathcal{S}'} Y = X$.



We use the following counting problem
(a counting version of the set cover problem).

Given X a finite set, $\mathcal{S} \subseteq 2^X$ a family

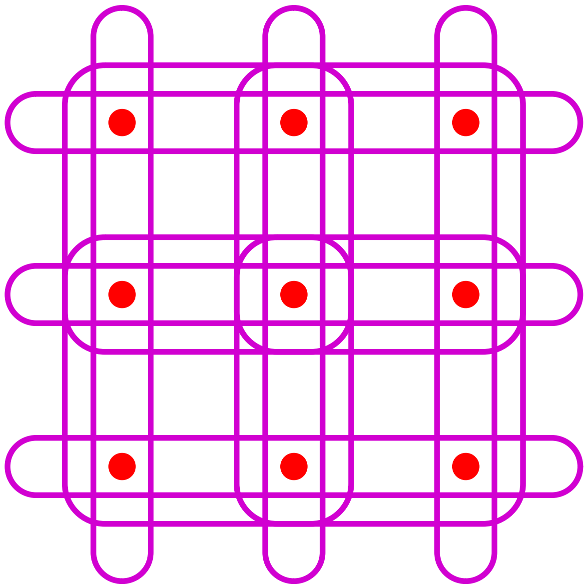
Output The number of subfamilies $\mathcal{S}' \subseteq \mathcal{S}$ st $\bigcup_{Y \in \mathcal{S}'} Y = X$.



We use the following counting problem
(a counting version of the set cover problem).

Given X a finite set, $\mathcal{S} \subseteq 2^X$ a family

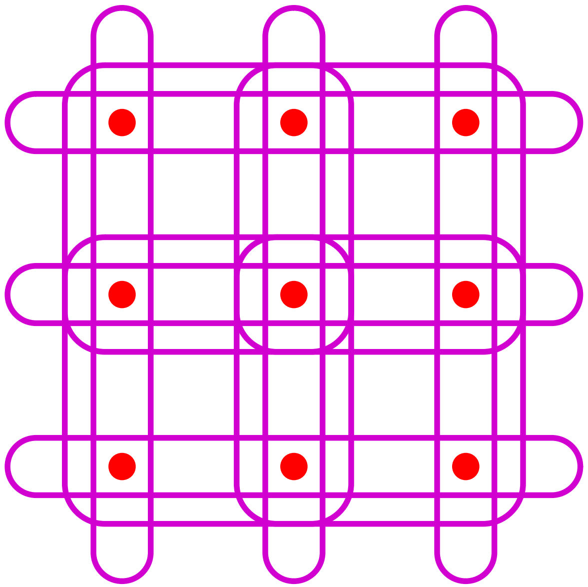
Output The number of subfamilies $\mathcal{S}' \subseteq \mathcal{S}$ st $\bigcup_{Y \in \mathcal{S}'} Y = X$.



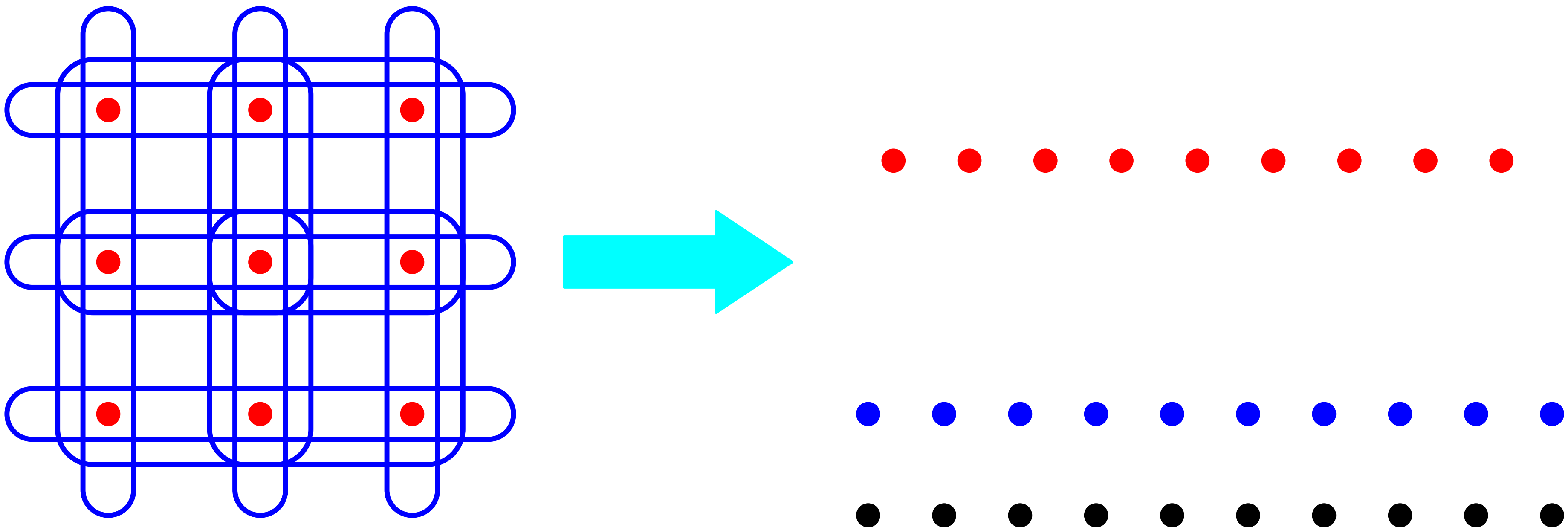
We use the following counting problem
(a counting version of the set cover problem).

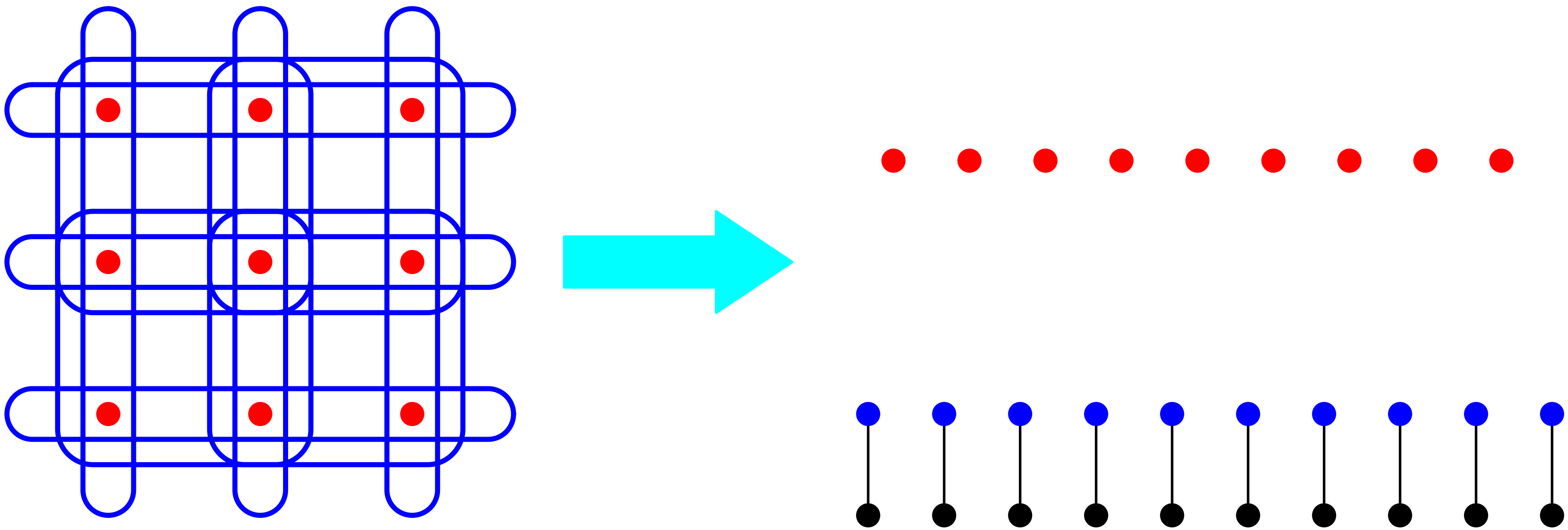
Given X a finite set, $\mathcal{S} \subseteq 2^X$ a family

Output The number of subfamilies $\mathcal{S}' \subseteq \mathcal{S}$ st $\bigcup_{Y \in \mathcal{S}'} Y = X$.

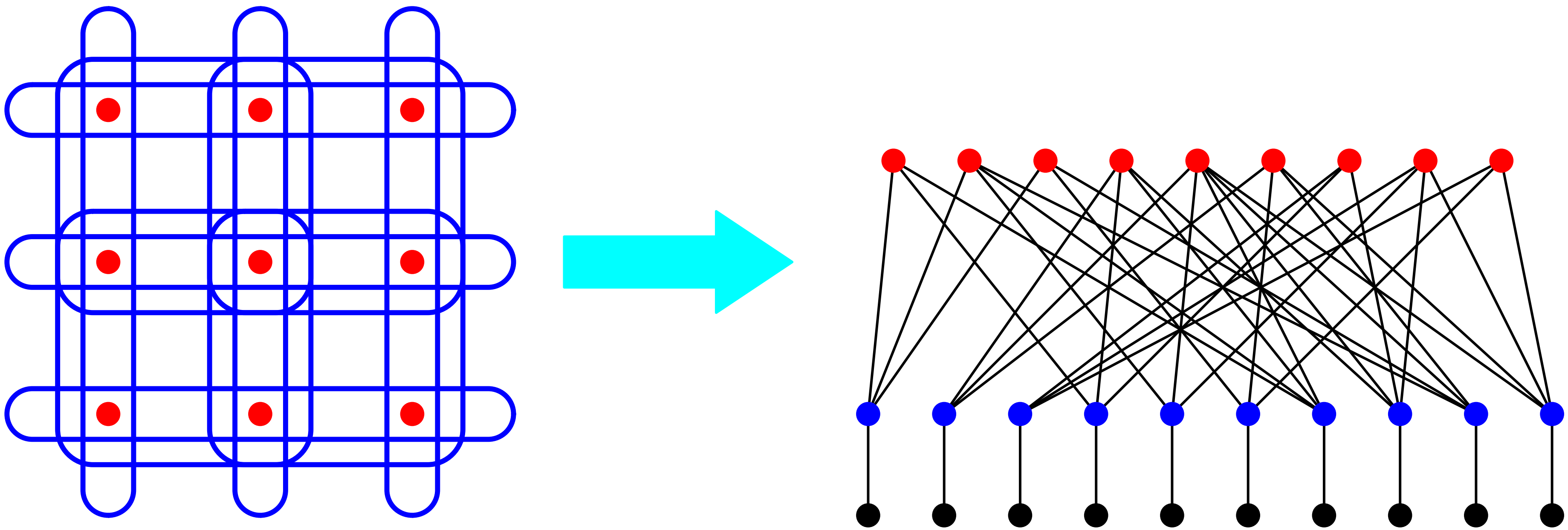


Actual reduction

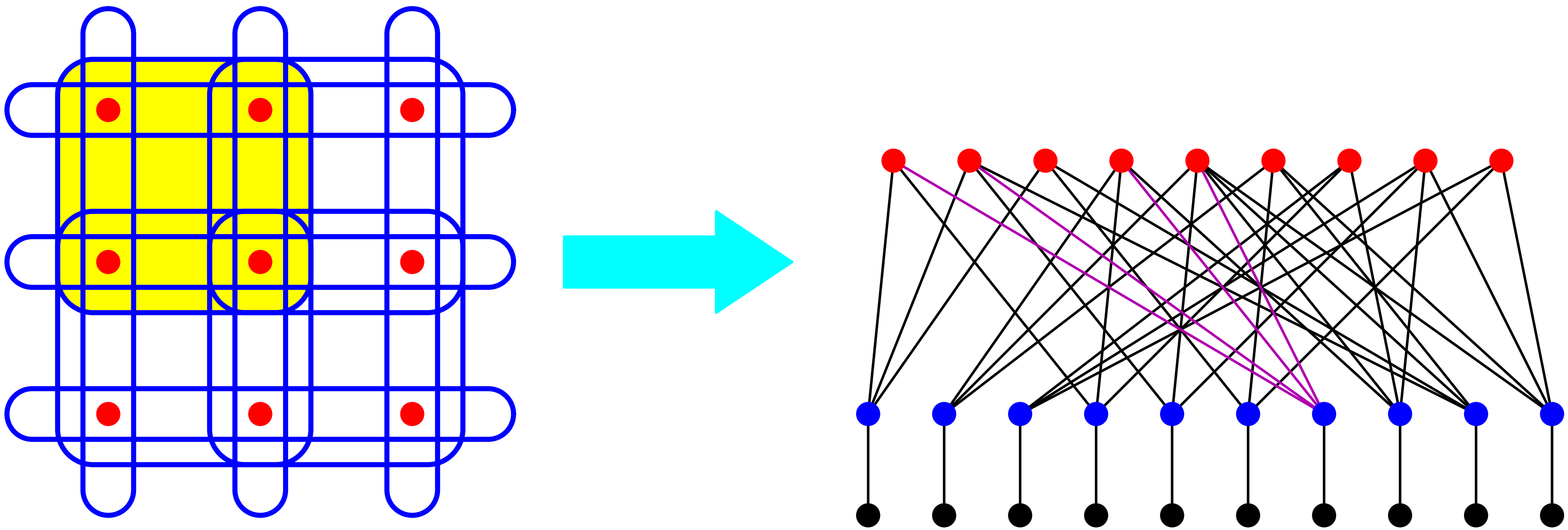


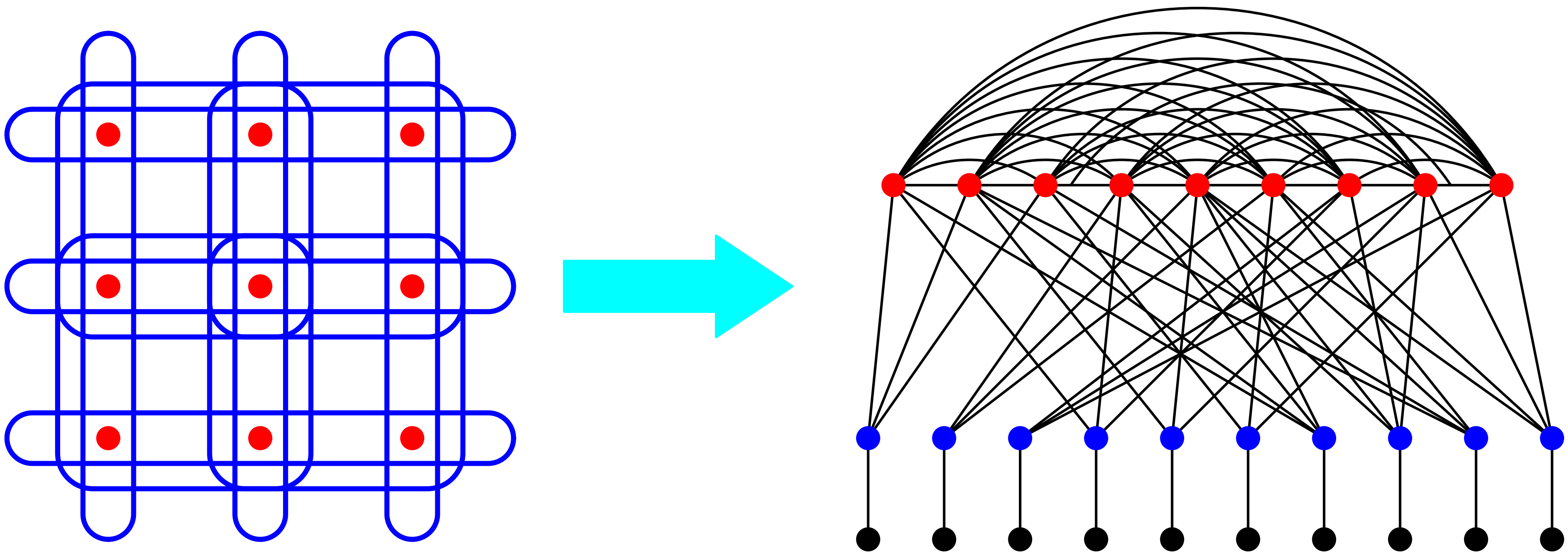


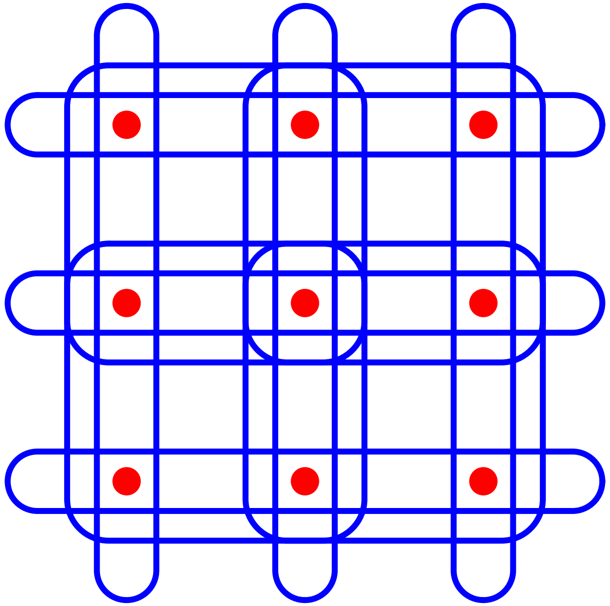
Actual reduction

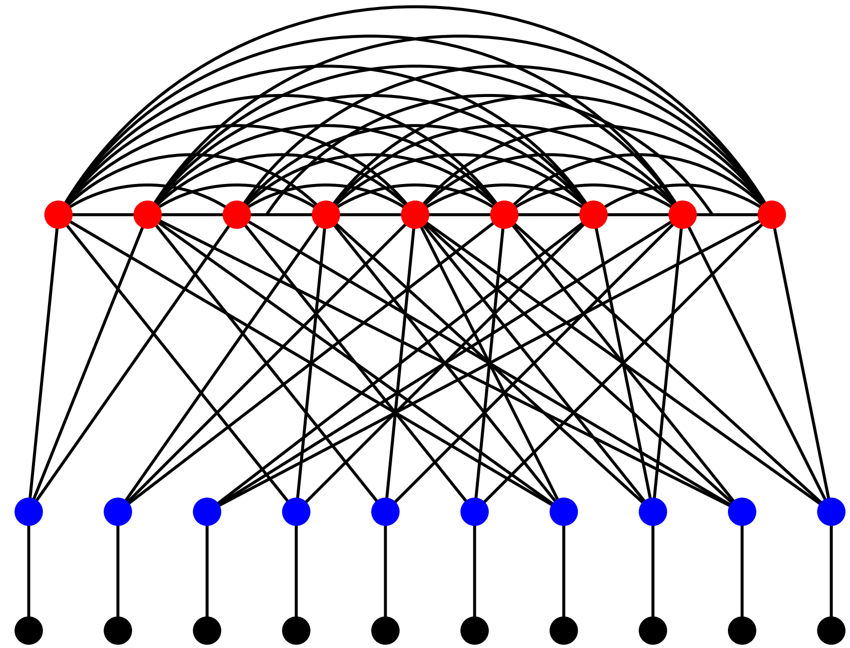


Actual reduction









$$\# \text{ set covers} + \sum_{x \in X} 2^{\#\text{ sets not containing } x}$$

$$\|$$

$$\# \text{ maximal independent sets}$$

We obtain the following results for chordal graphs.

(1) # independent sets of G
(2) # maximum independent sets of G
(3) # independent sets of G of fixed size

$O(|V| + |E|)$ alg.

(4) # maximal independent sets of G
(5) # minimum maximal independent sets of G

#P-complete

- “Enumeration” can be done in a similar way.
  - Yields algorithms with constant delay.
- Approximating a minimum-weight maximal independent set within a logarithmic factor is hard.

- "Enumeration" can be done in a similar way.
  - Yields algorithms with constant delay.
- Approximating a minimum-weight maximal independent set within a logarithmic factor is hard.

[End of my talk]

Merci beaucoup.