# Input-Sensitive Enumerations

Petr Golovach

Department of Informatics, University of Bergen

SGT 2018, Sète, France, 15.06.2018

# Plan of the lectures

- Introduction to branching enumeration algorithms and their analysis.
- Advanced analysis of branching algorithms; the "Measure and Conquer" technique.
- Lower bounds.
- Conclusions and open problems.

# Plan of the lectures

- Introduction to branching enumeration algorithms and their analysis.
- Advanced analysis of branching algorithms; the "Measure and Conquer" technique.
- Lower bounds.
- Conclusions and open problems.

# Upper and Lower bounds

Suppose that there is a family of instances $\mathcal{I}$ of an enumeration problem such that for every $n \in \mathbb{N}$, $\mathcal{I}$ contains an instance $I$ with $|I| = n$ and the number of enumerated objects for $I \in \mathcal{I}$ is $f(|I|)$.

Then $f(n)$ provides an unconditional *running time lower bound* for every enumeration algorithm.

# Upper and Lower bounds

Suppose that there is a family of instances $\mathcal{I}$ of an enumeration problem such that for every $n \in \mathbb{N}$, $\mathcal{I}$ contains an instance $I$ with $|I| = n$ and the number of enumerated objects for $I \in \mathcal{I}$ is $f(|I|)$.

Then $f(n)$ provides an unconditional *running time lower bound* for every enumeration algorithm.

Our aim is

- Construct an enumeration algorithm with the "best" running time.
- Construct the "best" lower bound.

# Upper and Lower bounds

Suppose that there is a family of instances $\mathcal{I}$ of an enumeration problem such that for every $n \in \mathbb{N}$, $\mathcal{I}$ contains an instance $I$ with $|I| = n$ and the number of enumerated objects for $I \in \mathcal{I}$ is $f(|I|)$.

Then $f(n)$ provides an unconditional *running time lower bound* for every enumeration algorithm.

Our aim is

- Construct an enumeration algorithm with the "best" running time.
- Construct the "best" lower bound.
- Ideally, we wish to get (asymptotically) tight upper and lower bounds for running time and combinatorial bounds for the number of enumerated objects.

# Upper and Lower bounds

Suppose that there is a family of instances $\mathcal{I}$ of an enumeration problem such that for every $n \in \mathbb{N}$, $\mathcal{I}$ contains an instance $I$ with $|I| = n$ and the number of enumerated objects for $I \in \mathcal{I}$ is $f(|I|)$.

Then $f(n)$ provides an unconditional *running time lower bound* for every enumeration algorithm.

Our aim is

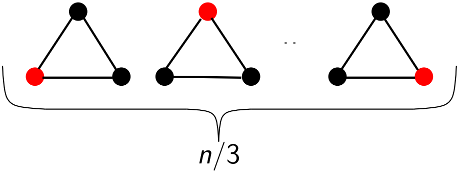
- Construct an enumeration algorithm with the "best" running time.
- Construct the "best" lower bound.
- Ideally, we wish to get (asymptotically) tight upper and lower bounds for running time and combinatorial bounds for the number of enumerated objects.
- If we fails to produce a lower bound that is "sufficiently close" to our upper bound, then this usually means that the upper bound is too big.

# Maximal independent sets

An $n$-vertex graph has at most $3^{n/3}$ maximal independent sets that can be enumerated in time $O^*(3^{n/3})$ and there are $n = 3k$-vertex graphs that have $3^{n/3}$ maximal independent sets.

# Maximal independent sets

An $n$-vertex graph has at most $3^{n/3}$ maximal independent sets that can be enumerated in time $O^*(3^{n/3})$ and there are $n = 3k$-vertex graphs that have $3^{n/3}$ maximal independent sets.

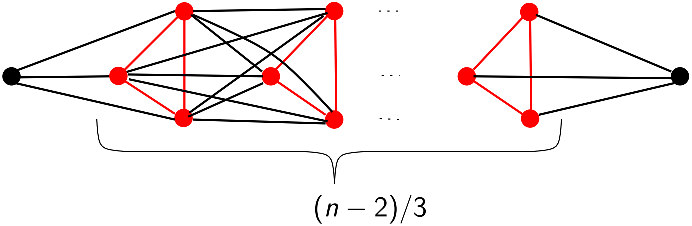


# Minimal connected dominating sets

An $n$-vertex chordal graph has at most $1.4736^n$ minimal connected dominating sets that can be enumerated in time $O(1.4736^n)$ and there are $n = 3k + 2$-vertex interval graphs that have $3^{(n-2)/3}$ $(1.44.22^n)$ minimal connected dominating sets.

# Minimal connected dominating sets

An $n$-vertex chordal graph has at most $1.4736^n$ minimal connected dominating sets that can be enumerated in time $O(1.4736^n)$ and there are $n = 3k + 2$-vertex interval graphs that have $3^{(n-2)/3}$ $(1.44.22^n)$ minimal connected dominating sets.

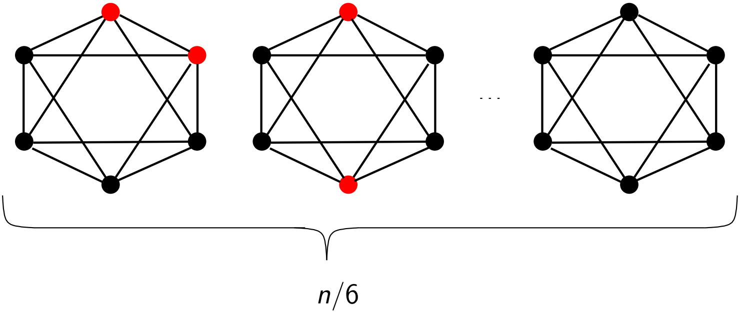


# Minimal dominating sets

An $n$-vertex graph has at most $1.4736^n$ minimal connected dominating sets that can be enumerated in time $O(1.7159^n)$ and there are $n = 6k$-vertex i graphs that have $15^{n/6}$ $(1.5704^n)$ minimal dominating sets.

# Minimal dominating sets

An $n$-vertex graph has at most $1.4736^n$ minimal connected dominating sets that can be enumerated in time $O(1.7159^n)$ and there are $n = 6k$-vertex i graphs that have $15^{n/6}$ $(1.5704^n)$ minimal dominating sets.

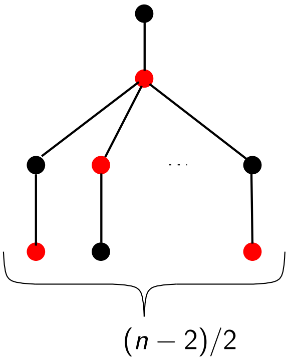


# Minimal dominating sets for trees

**Naive bound:** Three are $n = 2k$-vertex trees that have $2^{n/2}$ minimal dominating sets.

# Minimal dominating sets for trees

**Naive bound:** Three are $n = 2k$-vertex trees that have $2^{n/2}$ minimal dominating sets.



# Minimal dominating sets for trees

**Theorem (Krzywkowski, 2013)**

*There are $n = 27k$-vertex trees that have $12161^{n/27}$ minimal dominating sets.*

# Minimal dominating sets for trees

**Theorem (Krzywkowski, 2013)**

*There are $n = 27k$-vertex trees that have $12161^{n/27}$ minimal dominating sets.*

$$12161^{1/27} \approx 1.4167 > 2^{1/2} \approx 1.4142$$

# Minimal dominating sets for trees

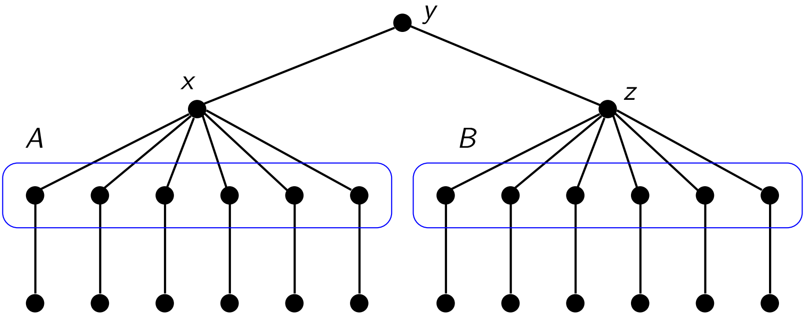
**Theorem (Krzywkowski, 2013)**

*There are $n = 27k$-vertex trees that have $12161^{n/27}$ minimal dominating sets.*

$$12161^{1/27} \approx 1.4167 > 2^{1/2} \approx 1.4142$$

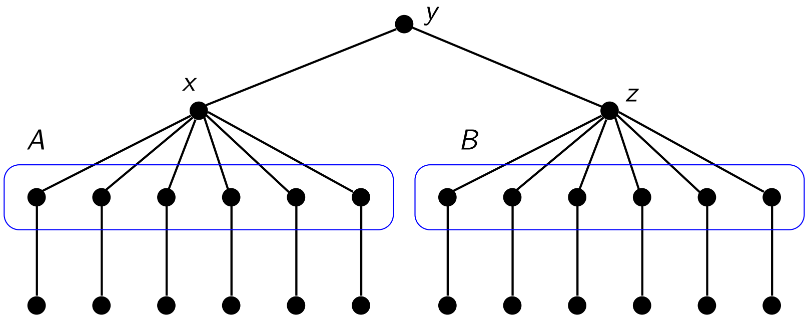
**Current upper bound:** $3^{n/3}$.

# Proof



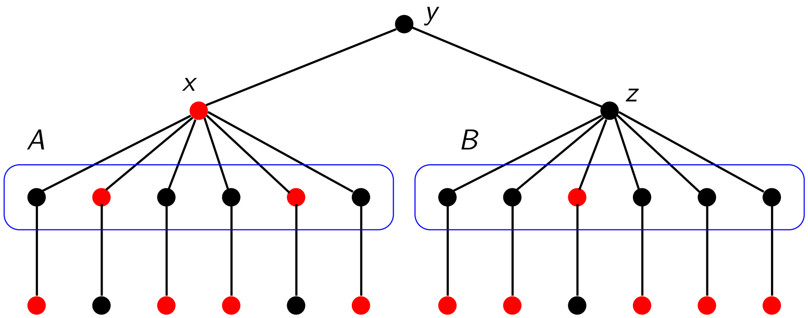
# Proof

There are $3 \cdot (2^6 - 1)^2$ minimal dominating sets $D$ such that $D \cap A \neq \emptyset$ and $D \cap B \neq \emptyset$.



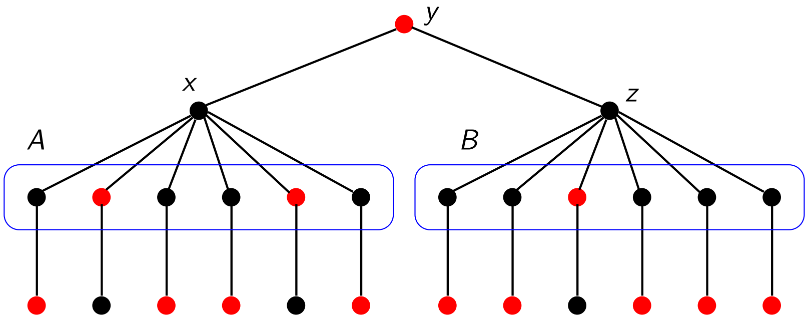
# Proof

There are $3 \cdot (2^6 - 1)^2$ minimal dominating sets $D$ such that $D \cap A \neq \emptyset$ and $D \cap B \neq \emptyset$.



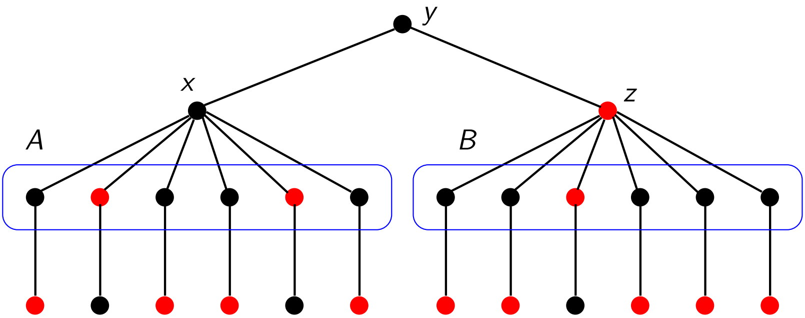
# Proof

There are $3 \cdot (2^6 - 1)^2$ minimal dominating sets $D$ such that $D \cap A \neq \emptyset$ and $D \cap B \neq \emptyset$.



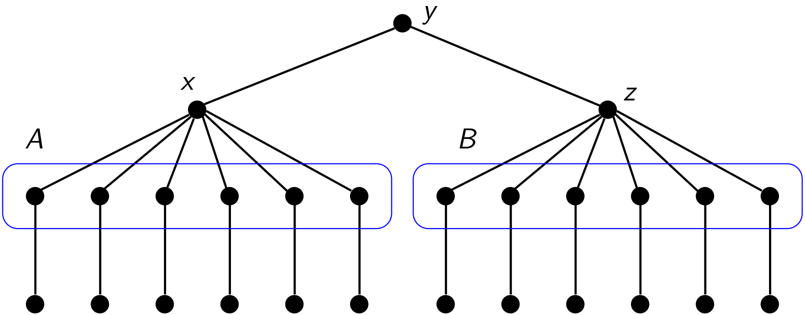
# Proof

There are $3 \cdot (2^6 - 1)^2$ minimal dominating sets $D$ such that $D \cap A \neq \emptyset$ and $D \cap B \neq \emptyset$.



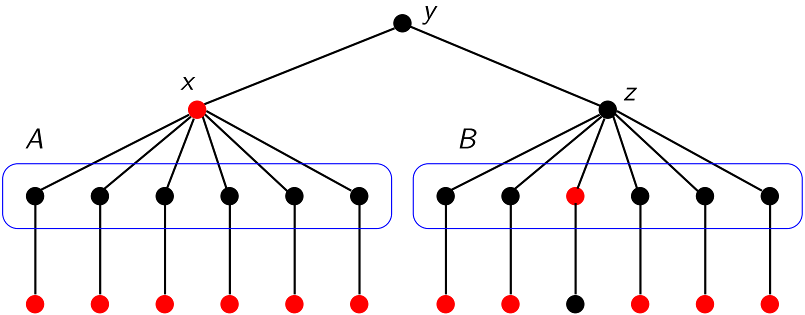
# Proof

There are $2 \cdot (2^6 - 1)$ minimal dominating sets $D$ such that $D \cap A = \emptyset$ and $D \cap B \neq \emptyset$ and, symmetrically, there are $2 \cdot (2^6 - 1)$ minimal dominating sets $D$ such that $D \cap A \neq \emptyset$ and $D \cap B = \emptyset$



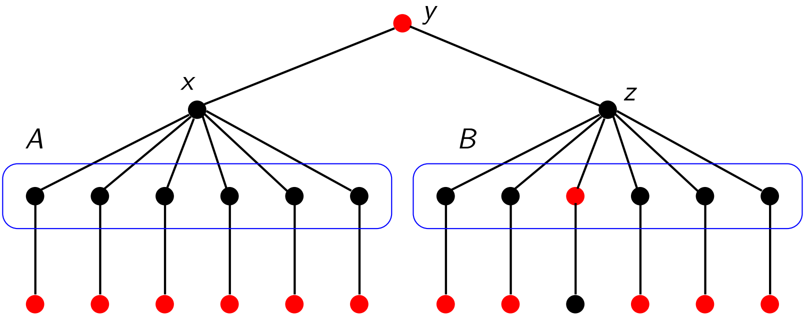
# Proof

There are $2 \cdot (2^6 - 1)$ minimal dominating sets $D$ such that $D \cap A = \emptyset$ and $D \cap B \neq \emptyset$ and, symmetrically, there are $2 \cdot (2^6 - 1)$ minimal dominating sets $D$ such that $D \cap A \neq \emptyset$ and $D \cap B = \emptyset$



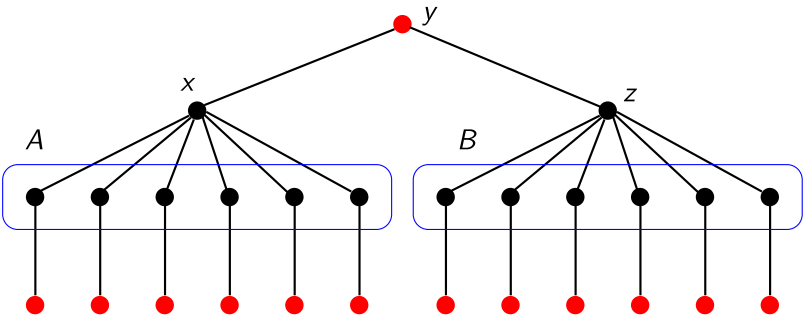
# Proof

There are $2 \cdot (2^6 - 1)$ minimal dominating sets $D$ such that $D \cap A = \emptyset$ and $D \cap B \neq \emptyset$ and, symmetrically, there are $2 \cdot (2^6 - 1)$ minimal dominating sets $D$ such that $D \cap A \neq \emptyset$ and $D \cap B = \emptyset$



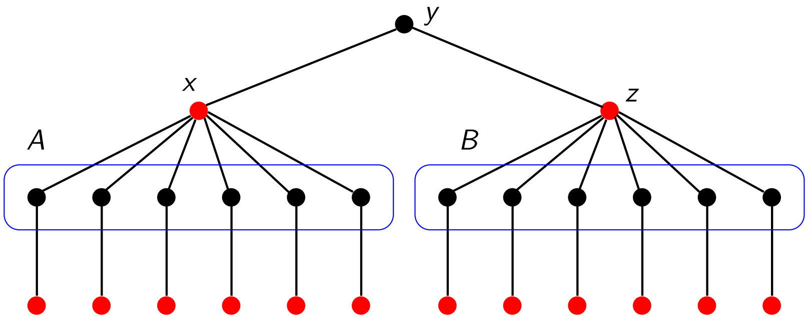
# Proof

There are 2 minimal dominating sets $D$ such that $D \cap A = \emptyset$ and $D \cap B = \emptyset$.



# Proof

There are 2 minimal dominating sets $D$ such that $D \cap A = \emptyset$ and $D \cap B = \emptyset$.



# Proof

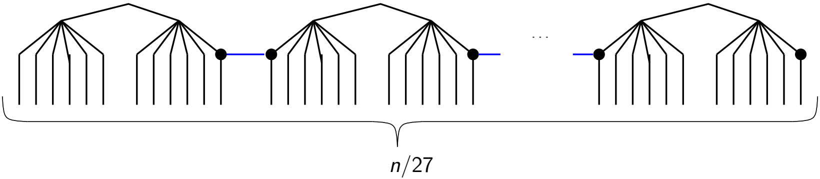
The total number of minimal dominating sets is

$$3 \cdot (2^6 - 1)^2 + 2 \cdot 2 \cdot (2^6 - 1) + 2 = 12161.$$

# Proof

The total number of minimal dominating sets is

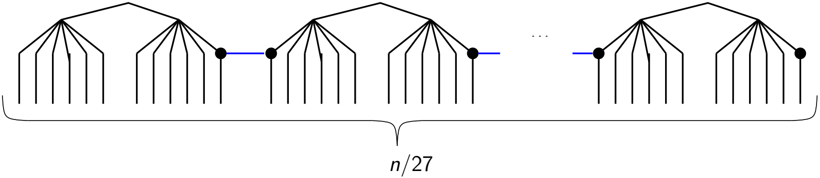
$$3 \cdot (2^6 - 1)^2 + 2 \cdot 2 \cdot (2^6 - 1) + 2 = 12161.$$



# Proof

The total number of minimal dominating sets is

$$3 \cdot (2^6 - 1)^2 + 2 \cdot 2 \cdot (2^6 - 1) + 2 = 12161.$$




This tree has $12161^{n/27}$ minimal dominating sets.

# Minimal separators

Let $G$ be a graph, and let $s$ and $t$ be distinct vertices of $G$.

# Minimal separators

Let $G$ be a graph, and let $s$ and $t$ be distinct vertices of $G$.

A set of vertices $S \subseteq V(G) \setminus \{s, t\}$ is an *$(s, t)$-separator* if $s$ and $t$ are in distinct components of $G - \{s, t\}$.

# Minimal separators

Let $G$ be a graph, and let $s$ and $t$ be distinct vertices of $G$.

A set of vertices $S \subseteq V(G) \setminus \{s, t\}$ is an *$(s, t)$-separator* if $s$ and $t$ are in distinct components of $G - \{s, t\}$.

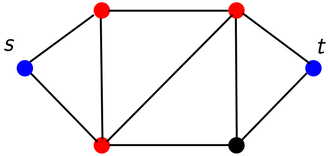
An $(s, t)$-separator is *(inclusion) minimal* if every $S' \subset S$ is not an $(s, t)$-separator.

# Minimal separators

Let $G$ be a graph, and let $s$ and $t$ be distinct vertices of $G$.

A set of vertices $S \subseteq V(G) \setminus \{s, t\}$ is an *$(s, t)$-separator* if $s$ and $t$ are in distinct components of $G - \{s, t\}$.

An $(s, t)$-separator is *(inclusion) minimal* if every $S' \subset S$ is not an $(s, t)$-separator.

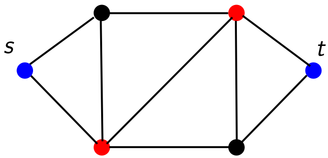


# Minimal separators

Let $G$ be a graph, and let $s$ and $t$ be distinct vertices of $G$.

A set of vertices $S \subseteq V(G) \setminus \{s, t\}$ is an *$(s, t)$-separator* if $s$ and $t$ are in distinct components of $G - \{s, t\}$.

An $(s, t)$-separator is *(inclusion) minimal* if every $S' \subset S$ is not an $(s, t)$-separator.

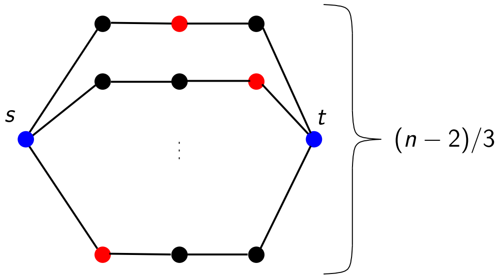


# Minimal separators

**Naive bound:** Three are $n = 3k + 2$-vertex graphs that have $3^{(n-2)/3}$ minimal separators.

# Minimal separators

**Naive bound:** Three are $n = 3k + 2$-vertex graphs that have $3^{(n-2)/3}$ minimal separators.



# Minimal separators

**Theorem (Gaspers and Mackenzie, 2017)**

*There are $n$-vertex graphs that have at least $1.4457^n$ minimal $(s,t)$-separators.*

# Minimal separators

**Theorem (Gaspers and Mackenzie, 2017)**

*There are n-vertex graphs that have at least* $1.4457^n$ *minimal* $(s,t)$*-separators.*

$1.4457 > 3^{1/3} \approx 1.4422.$

## Minimal separators

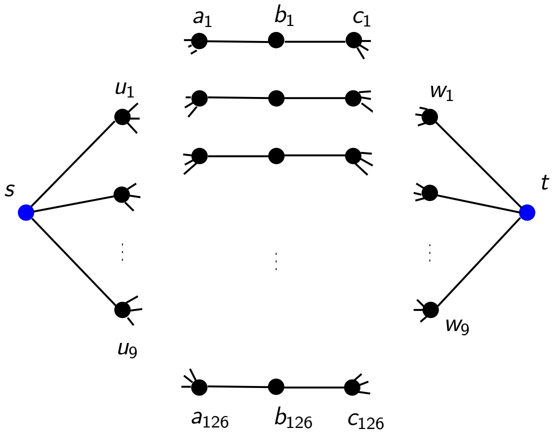
**Theorem (Gaspers and Mackenzie, 2017)**

*There are n-vertex graphs that have at least* $1.4457^n$ *minimal* $(s,t)$*-separators.*

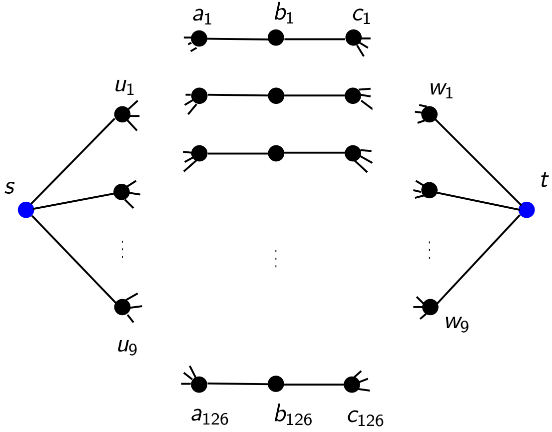
$1.4457 > 3^{1/3} \approx 1.4422.$

**Current upper bound:** $O(1.6180^n)$.

# Sketch of the proof

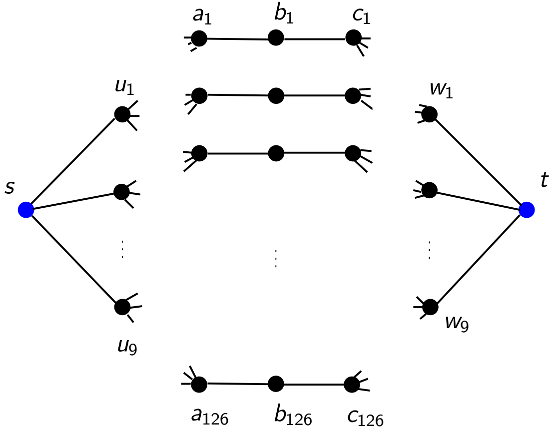


# Sketch of the proof



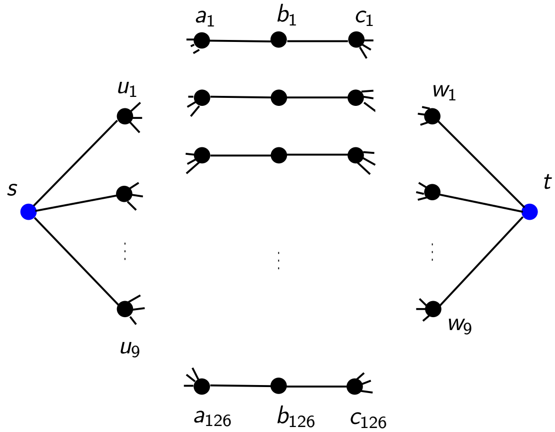

Note that $126 = \binom{9}{4}$; we make each $a_i$ adjacent to 4 vertices of $\{u_1, \ldots, u_9\}$ in such a way that $a_i$-s have distinct neighborhoods.

# Sketch of the proof



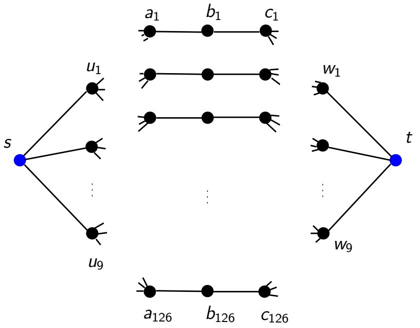

Symmetrically we make each $c_i$ adjacent to 4 vertices of $\{w_1, \ldots, w_9\}$ in such a way that $c_i$-s have distinct neighborhoods.

# Sketch of the proof



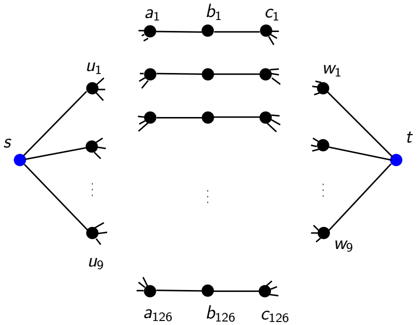

**Claim:** This graph has $> 2.4603 \cdot 10^{63}$ minimal $(s, t)$-separators.

# Sketch of the proof




For $0 \leq p, q \leq 9$, let $N_{s,t}$ be the number of minimal $(s,t)$-separators that have $p$ vertices from $\{u_1, \ldots, u_9\}$ and $q$ vertices from $\{w_1, \ldots, w_9\}$.

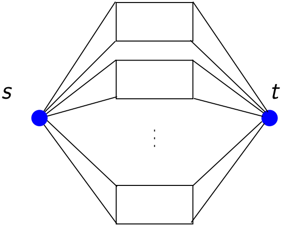
# Sketch of the proof



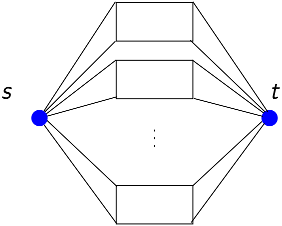

For $0 \leq p, q \leq 9$, let $N_{s,t}$ be the number of minimal $(s,t)$-separators that have $p$ vertices from $\{u_1, \ldots, u_9\}$ and $q$ vertices from $\{w_1, \ldots, w_9\}$.

Consider the cases (i) $p, q = 0$, (ii) $p = 0$ and $q \geq 4$, (iii) $p \geq 4$ and $q = 0$, (iv) $p, q \geq 4$ and lower bound $|S_{p,q}|$.

# Sketch of the proof



# Sketch of the proof




This graph has at least $1.4457^n$ minimal $(s, t)$-separators.

# Input-sensitive enumeration

- The running time depends on the length of the input only (e.g., the number of vertices of the input graph).
- We use the classical worst case running time analysis.
- If the number of objects to be enumerated is *exponential* (in the worst case), then an input-sensitive enumeration algorithm runs in *exponential* time.
- We use *exact exponential-time algorithms*, in particular *branching algorithms*.

# Input-sensitive enumeration

We considered

- basics of branching enumeration algorithms and their analysis,

# Input-sensitive enumeration

We considered

- basics of branching enumeration algorithms and their analysis,
- advanced analysis of branching algorithms using the "Measure & Conquer" technique,

# Input-sensitive enumeration

We considered

- basics of branching enumeration algorithms and their analysis,
- advanced analysis of branching algorithms using the "Measure & Conquer" technique,
- lower bounds.

# Input-sensitive enumeration

We considered

- basics of branching enumeration algorithms and their analysis,
- advanced analysis of branching algorithms using the "Measure & Conquer" technique,
- lower bounds.

Other techniques:

# Input-sensitive enumeration

We considered

- basics of branching enumeration algorithms and their analysis,
- advanced analysis of branching algorithms using the "Measure & Conquer" technique,
- lower bounds.

Other techniques:

- Brute force.

# Input-sensitive enumeration

We considered

- basics of branching enumeration algorithms and their analysis,
- advanced analysis of branching algorithms using the "Measure & Conquer" technique,
- lower bounds.

Other techniques:

- Brute force.
- Dynamic programming.

# Input-sensitive enumeration

We considered

- basics of branching enumeration algorithms and their analysis,
- advanced analysis of branching algorithms using the "Measure & Conquer" technique,
- lower bounds.

Other techniques:

- Brute force.
- Dynamic programming.
- ...

# Input-sensitive enumeration

We considered

- basics of branching enumeration algorithms and their analysis,
- advanced analysis of branching algorithms using the “Measure & Conquer” technique,
- lower bounds.

Other techniques:

- Brute force.
- Dynamic programming.
- . . .
- Combinations of distinct techniques.

# Schöning's algorithm for 3-Satisfiability

**Problem (3-Satisfiability)**

**Input:** *A Boolean formula $\phi$ with $n$ variables in the conjunctive normal form such that each clause contain 3 literals.*

**Task:** *Decide whether $\phi$ has a satisfying assignment of variables.*

# Schöning's algorithm for 3-Satisfiability

**Problem (3-Satisfiability)**

**Input:** *A Boolean formula $\phi$ with $n$ variables in the conjunctive normal form such that each clause contain 3 literals.*

**Task:** *Decide whether $\phi$ has a satisfying assignment of variables.*

$$\phi = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_3)$$

# Schöning's algorithm for 3-Satisfiability

The algorithm (Schöning, 1999):

# Schöning's algorithm for 3-Satisfiability

The algorithm (Schöning, 1999):

- Assign the values of of the variables uniformly at random.

# Schöning's algorithm for 3-Satisfiability

The algorithm (Schöning, 1999):

- Assign the values of of the variables uniformly at random.
- Search for a satisfying assignment at Hamming distance at most $n/4$.

# Schöning's algorithm for 3-Satisfiability

The algorithm (Schöning, 1999):

- Assign the values of of the variables uniformly at random.
- Search for a satisfying assignment at Hamming distance at most $n/4$.

The algorithm runs in time $O^*(1.5^n)$.

# Parameterized complexity

*Parameterized Complexity* is a two-dimensional framework for studying the computations complexity;

- one dimension is the input size $n$,
- another one is a *parameter* $k$.

# Parameterized complexity

*Parameterized Complexity* is a two-dimensional framework for studying the computations complexity;

- one dimension is the input size $n$,
- another one is a *parameter* $k$.

A parameterized problem is *fixed-parameter tractable (FPT)* if it can be solved in time

$$f(k) \cdot n^{O(1)}.$$

# Extension problems

Let $P$ be a property of vertex subsets of a graph.

# Extension problems

Let $P$ be a property of vertex subsets of a graph.
The property can depend on the input graph but we should be able to test whether a set satisfies the property in polynomial time.

# Extension problems

Let $P$ be a property of vertex subsets of a graph.
The property can depend on the input graph but we should be able to test whether a set satisfies the property in polynomial time.

**Problem ($P$-Subset)**

**Input:** *A graph $G$.*

**Task:** *Decide whether there is a set $U \subseteq V(G)$ satisfying $P$.*

# Extension problems

Let $P$ be a property of vertex subsets of a graph.
The property can depend on the input graph but we should be able to test whether a set satisfies the property in polynomial time.

**Problem ($P$-Subset)**

**Input:** *A graph $G$.*
**Task:** *Decide whether there is a set $U \subseteq V(G)$ satisfying $P$.*

**Problem ($P$-Extension)**

**Input:** *A graph $G$, $U \subseteq V(G)$, and a non-negative integer $k$.*
**Parameter:** $k$
**Task:** *Decide whether there is a set $X \subseteq V(G) \setminus U$ of size at most $k$ such that $U \cup X$ satisfies $P$.*

# Exact algorithm via Local Search

Fedor V. Fomin, Serge Gaspers, Daniel Lokshtanov, Saket Saurabh: Exact algorithms via monotone local search. STOC 2016: 764-775.

# Exact algorithm via Local Search

Fedor V. Fomin, Serge Gaspers, Daniel Lokshtanov, Saket Saurabh: Exact algorithms via monotone local search. STOC 2016: 764-775.

**Theorem**

*If there exists an algorithm for P-Extension with running time $c^k n^{O(1)}$, then there exists a randomized algorithm for P-Subset with running time $O^*((2 - \frac{1}{c})^n)$.*

# Exact algorithm via Local Search

Fedor V. Fomin, Serge Gaspers, Daniel Lokshtanov, Saket Saurabh: Exact algorithms via monotone local search. STOC 2016: 764-775.

**Theorem**

*If there exists an algorithm for P-Extension with running time $c^k n^{O(1)}$, then there exists a randomized algorithm for P-Subset with running time $O^*((2 - \frac{1}{c})^n)$.*

**Theorem**

*If there exists an algorithm for P-Extension with running time $c^k n^{O(1)}$, then there exists a deterministic algorithm for P-Subset with running time $O^*((2 - \frac{1}{c})^{n+o(n)})$.*

# Minimal feedback vertex sets

A set of vertices $X$ of a graph $G$ is a *feedback vertex set* if $G - X$ is acyclic.

# Minimal feedback vertex sets

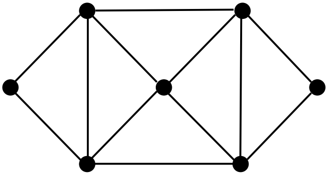
A set of vertices $X$ of a graph $G$ is a *feedback vertex set* if $G - X$ is acyclic.

A feedback vertex set is *(inclusion) minimal* if every $Y \subset X$ is not a feedback vertex set.

# Minimal feedback vertex sets

A set of vertices $X$ of a graph $G$ is a *feedback vertex set* if $G - X$ is acyclic.

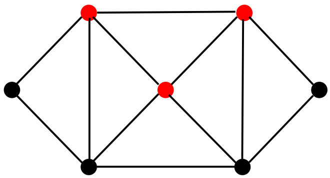
A feedback vertex set is *(inclusion) minimal* if every $Y \subset X$ is not a feedback vertex set.



# Minimal feedback vertex sets

A set of vertices $X$ of a graph $G$ is a *feedback vertex set* if $G - X$ is acyclic.

A feedback vertex set is *(inclusion) minimal* if every $Y \subset X$ is not a feedback vertex set.



# Minimal feedback vertex sets

**Theorem (Fomin, Gaspers, Pyatkin, Razgon, 2008)**

*An $n$-vertex graph has at most $1.8638^n$ minimal feedback vertex sets and these sets can be enumerated in time $O(1.8638^n)$.*

# Minimal feedback vertex sets

**Theorem (Fomin, Gaspers, Pyatkin, Razgon, 2008)**

*An $n$-vertex graph has at most $1.8638^n$ minimal feedback vertex sets and these sets can be enumerated in time $O(1.8638^n)$.*

**Theorem (Gaspers and Lee, 2017)**

*An $n$-vertex graph has $O(1.8527^n)$ minimal feedback vertex sets and these sets can be enumerated in time $O(1.8527^n)$.*

# Minimal feedback vertex sets

**Theorem (Fomin, Gaspers, Pyatkin, Razgon, 2008)**

*An $n$-vertex graph has at most $1.8638^n$ minimal feedback vertex sets and these sets can be enumerated in time $O(1.8638^n)$.*

**Theorem (Gaspers and Lee, 2017)**

*An $n$-vertex graph has $O(1.8527^n)$ minimal feedback vertex sets and these sets can be enumerated in time $O(1.8527^n)$.*

Lower bound: There are $n = 10k$-vertex graphs with at least $105^{n/10}$ ($1.5926^n$) minimal feedback vertex sets.

# Enumeration of minimal hitting sets

Let $\mathcal{S}$ be a family of subsets over an universe $\mathcal{U}$.

# Enumeration of minimal hitting sets

Let $\mathcal{S}$ be a family of subsets over an universe $\mathcal{U}$.

$X \subseteq \mathcal{U}$ is a hitting set for $\mathcal{S}$ if for every $S \in \mathcal{S}$, $X \cap S \neq \emptyset$.

# Enumeration of minimal hitting sets

Let $\mathcal{S}$ be a family of subsets over an universe $\mathcal{U}$.

$X \subseteq \mathcal{U}$ is a hitting set for $\mathcal{S}$ if for every $S \in \mathcal{S}$, $X \cap S \neq \emptyset$.

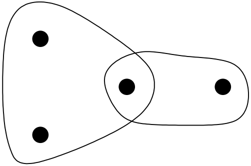
A hitting set $X$ is (inclusion) minimal if every $Y \subset X$ is not a hitting set.

# Enumeration of minimal hitting sets

Let $\mathcal{S}$ be a family of subsets over an universe $\mathcal{U}$.

$X \subseteq \mathcal{U}$ is a hitting set for $\mathcal{S}$ if for every $S \in \mathcal{S}$, $X \cap S \neq \emptyset$.

A hitting set $X$ is (inclusion) minimal if every $Y \subset X$ is not a hitting set.

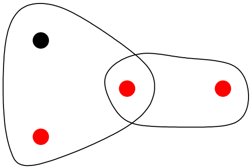


# Enumeration of minimal hitting sets

Let $\mathcal{S}$ be a family of subsets over an universe $\mathcal{U}$.

$X \subseteq \mathcal{U}$ is a hitting set for $\mathcal{S}$ if for every $S \in \mathcal{S}$, $X \cap S \neq \emptyset$.

A hitting set $X$ is (inclusion) minimal if every $Y \subset X$ is not a hitting set.

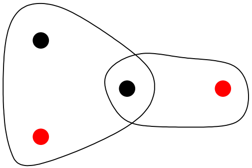


# Enumeration of minimal hitting sets

Let $\mathcal{S}$ be a family of subsets over an universe $\mathcal{U}$.

$X \subseteq \mathcal{U}$ is a hitting set for $\mathcal{S}$ if for every $S \in \mathcal{S}$, $X \cap S \neq \emptyset$.

A hitting set $X$ is (inclusion) minimal if every $Y \subset X$ is not a hitting set.



# Enumeration of minimal hitting sets

We proved that if $\mathcal{S}$ contains sets of size at most 3, then $\mathcal{S}$ has at most $1.8394^n$ minimal hitting sets and these sets can be enumerated it time $O(1.8394^n)$ where $n = |\mathcal{U}|$.

# Enumeration of minimal hitting sets

We proved that if $\mathcal{S}$ contains sets of size at most 3, then $\mathcal{S}$ has at most $1.8394^n$ minimal hitting sets and these sets can be enumerated it time $O(1.8394^n)$ where $n = |\mathcal{U}|$.

**Theorem (Cochefert, Couturier, Kratsch, 2016)**

*If $\mathcal{S}$ contains sets of size at most 3, then minimal hitting sets of $\mathcal{S}$ can be enumerated it time $O(1.6755^n)$ where $n = |\mathcal{U}|$.*

# Enumeration of minimal hitting sets

We proved that if $\mathcal{S}$ contains sets of size at most 3, then $\mathcal{S}$ has at most $1.8394^n$ minimal hitting sets and these sets can be enumerated it time $O(1.8394^n)$ where $n = |\mathcal{U}|$.

**Theorem (Cochefert, Couturier, Kratsch, 2016)**

*If $\mathcal{S}$ contains sets of size at most 3, then minimal hitting sets of $\mathcal{S}$ can be enumerated it time $O(1.6755^n)$ where $n = |\mathcal{U}|$.*

**Theorem (Gaspers and Lee, 2017)**

*If $\mathcal{S}$ contains sets of size at most 3, then minimal hitting sets of $\mathcal{S}$ can be enumerated it time $O(1.6727^n)$ where $n = |\mathcal{U}|$.*

# Enumeration of minimal hitting sets

We proved that if $\mathcal{S}$ contains sets of size at most 3, then $\mathcal{S}$ has at most $1.8394^n$ minimal hitting sets and these sets can be enumerated it time $O(1.8394^n)$ where $n = |\mathcal{U}|$.

**Theorem (Cochefert, Couturier, Kratsch, 2016)**

*If $\mathcal{S}$ contains sets of size at most 3, then minimal hitting sets of $\mathcal{S}$ can be enumerated it time $O(1.6755^n)$ where $n = |\mathcal{U}|$.*

**Theorem (Gaspers and Lee, 2017)**

*If $\mathcal{S}$ contains sets of size at most 3, then minimal hitting sets of $\mathcal{S}$ can be enumerated it time $O(1.6727^n)$ where $n = |\mathcal{U}|$.*

Lower bound: There is a family of sets $\mathcal{S}$ that have $1.5848^n$ minimal hitting sets.

# Limitations of branching algorithms

All efficient branching algorithms for subsets enumerations are using *local* properties.

# Limitations of branching algorithms

All efficient branching algorithms for subsets enumerations are using *local* properties.

For example,

# Limitations of branching algorithms

All efficient branching algorithms for subsets enumerations are using *local* properties.

For example, a set of vertices $X \subseteq V(G)$ is a maximal independent set if and only if

- for every $v \in X$, the neighbors of $v$ are not in $X$,

# Limitations of branching algorithms

All efficient branching algorithms for subsets enumerations are using *local* properties.

For example, a set of vertices $X \subseteq V(G)$ is a maximal independent set if and only if

- for every $v \in X$, the neighbors of $v$ are not in $X$,
- for every $u \in V(G)$, at least one neighbor of $u$ is in $X$.

# Limitations of branching algorithms

All efficient branching algorithms for subsets enumerations are using *local* properties.

For example, a set of vertices $X \subseteq V(G)$ is a maximal independent set if and only if

- for every $v \in X$, the neighbors of $v$ are not in $X$,
- for every $u \in V(G)$, at least one neighbor of $u$ is in $X$.

**Task:** Develop enumeration techniques for sets defined by *non-local* properties.

# Enumeration of connected dominating sets

A set of vertices $D$ of a graph $G$ is a *connected dominating* set if

- $D$ is a dominating set,

# Enumeration of connected dominating sets

A set of vertices $D$ of a graph $G$ is a *connected dominating* set if

- $D$ is a dominating set,
- $G[D]$ is connected.

# Enumeration of connected dominating sets

A set of vertices $D$ of a graph $G$ is a *connected dominating* set if

- $D$ is a dominating set,
- $G[D]$ is connected.

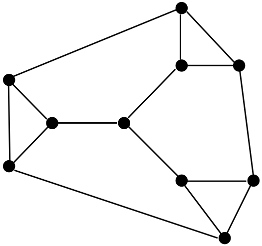
A connected dominating set $D$ is *minimal* if $D$ is a connected dominating set and for every $D' \subset D$, $D'$ is not a connected dominating set.

# Enumeration of connected dominating sets

A set of vertices $D$ of a graph $G$ is a *connected dominating* set if

- $D$ is a dominating set,
- $G[D]$ is connected.

A connected dominating set $D$ is *minimal* if $D$ is a connected dominating set and for every $D' \subset D$, $D'$ is not a connected dominating set.

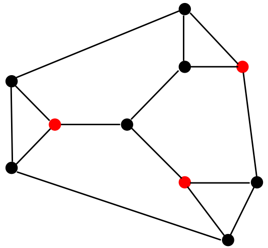


# Enumeration of connected dominating sets

A set of vertices $D$ of a graph $G$ is a *connected dominating* set if

- $D$ is a dominating set,
- $G[D]$ is connected.

A connected dominating set $D$ is *minimal* if $D$ is a connected dominating set and for every $D' \subset D$, $D'$ is not a connected dominating set.

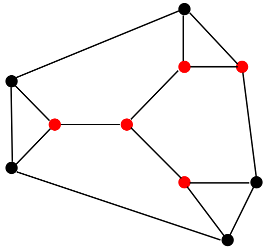


# Enumeration of connected dominating sets

A set of vertices $D$ of a graph $G$ is a *connected dominating* set if

- $D$ is a dominating set,
- $G[D]$ is connected.

A connected dominating set $D$ is *minimal* if $D$ is a connected dominating set and for every $D' \subset D$, $D'$ is not a connected dominating set.

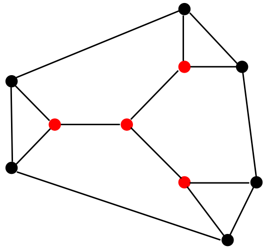


# Enumeration of connected dominating sets

A set of vertices $D$ of a graph $G$ is a *connected dominating* set if

- $D$ is a dominating set,
- $G[D]$ is connected.

A connected dominating set $D$ is *minimal* if $D$ is a connected dominating set and for every $D' \subset D$, $D'$ is not a connected dominating set.



# Enumeration of connected dominating sets

## Problem (Enumeration of Minimal CDS)

**Input:** *A connected graph $G$.*

**Task:** *Enumerate all minimal connected dominated sets.*

# Enumeration of connected dominating sets

## Problem (Enumeration of Minimal CDS)

**Input:** *A connected graph $G$.*

**Task:** *Enumerate all minimal connected dominated sets.*

All minimal connected dominating sets of an $n$ vertex graph can be enumerated in time $O^*(2^n)$.

# Enumeration of connected dominating sets

## Problem (Enumeration of Minimal CDS)

**Input:** *A connected graph $G$.*

**Task:** *Enumerate all minimal connected dominated sets.*

All minimal connected dominating sets of an $n$ vertex graph can be enumerated in time $O^*(2^n)$.

All minimal connected dominating sets can be enumerated in time $O^*(2^{(1-\varepsilon)n})$ for some (small) $\varepsilon > 0$ (Lokshtanov, Pilipczuk, Saurabh, 2016).

# Enumeration of connected dominating sets

## Problem (Enumeration of Minimal CDS)

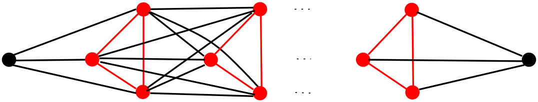
**Input:** *A connected graph $G$.*

**Task:** *Enumerate all minimal connected dominated sets.*

All minimal connected dominating sets of an $n$ vertex graph can be enumerated in time $O^*(2^n)$.

All minimal connected dominating sets can be enumerated in time $O^*(2^{(1-\varepsilon)n})$ for some (small) $\varepsilon > 0$ (Lokshtanov, Pilipczuk, Saurabh, 2016).

There are graphs with at least $3^{(n-2)/3}$ minimal CDS:



# Enumeration of connected vertex covers

A set of vertices $X$ of a graph $G$ is a *connected vertex cover* if

- $X$ is a vertex cover, that is, for every $uv \in E(G)$, $u \in X$ or $v \in X$,

# Enumeration of connected vertex covers

A set of vertices $X$ of a graph $G$ is a *connected vertex cover* if

- $X$ is a vertex cover, that is, for every $uv \in E(G)$, $u \in X$ or $v \in X$,
- $G[X]$ is connected.

# Enumeration of connected vertex covers

A set of vertices $X$ of a graph $G$ is a *connected vertex cover* if

- $X$ is a vertex cover, that is, for every $uv \in E(G)$, $u \in X$ or $v \in X$,
- $G[X]$ is connected.

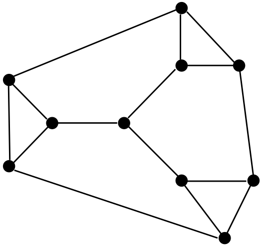
A connected vertex cover $X$ is *minimal* if $X$ is a connected vertex cover and for every $X' \subset X$, $X'$ is not a connected vertex cover.

# Enumeration of connected vertex covers

A set of vertices $X$ of a graph $G$ is a *connected vertex cover* if

- $X$ is a vertex cover, that is, for every $uv \in E(G)$, $u \in X$ or $v \in X$,
- $G[X]$ is connected.

A connected vertex cover $X$ is *minimal* if $X$ is a connected vertex cover and for every $X' \subset X$, $X'$ is not a connected vertex cover.

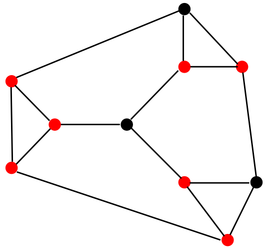


# Enumeration of connected vertex covers

A set of vertices $X$ of a graph $G$ is a *connected vertex cover* if

- $X$ is a vertex cover, that is, for every $uv \in E(G)$, $u \in X$ or $v \in X$,
- $G[X]$ is connected.

A connected vertex cover $X$ is *minimal* if $X$ is a connected vertex cover and for every $X' \subset X$, $X'$ is not a connected vertex cover.

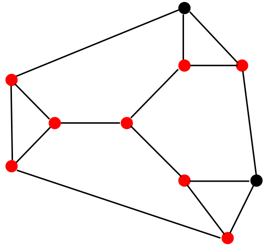


# Enumeration of connected vertex covers

A set of vertices $X$ of a graph $G$ is a *connected vertex cover* if

- $X$ is a vertex cover, that is, for every $uv \in E(G)$, $u \in X$ or $v \in X$,
- $G[X]$ is connected.

A connected vertex cover $X$ is *minimal* if $X$ is a connected vertex cover and for every $X' \subset X$, $X'$ is not a connected vertex cover.

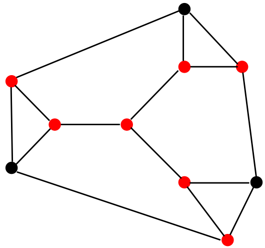


# Enumeration of connected vertex covers

A set of vertices $X$ of a graph $G$ is a connected vertex cover if

- $X$ is a vertex cover, that is, for every $uv \in E(G)$, $u \in X$ or $v \in X$,
- $G[X]$ is connected.

A connected vertex cover $X$ is *minimal* if $X$ is a connected vertex cover and for every $X' \subset X$, $X'$ is not a connected vertex cover.



# Enumeration of connected vertex covers

**Problem (Enumeration of Minimal CVC)**

**Input:** *A connected graph $G$.*

**Task:** *Enumerate all minimal connected vertex covers.*

# Enumeration of connected vertex covers

**Problem (Enumeration of Minimal CVC)**

**Input:** *A connected graph $G$.*

**Task:** *Enumerate all minimal connected vertex covers.*

An $n$-vertex graph has at most $2 \cdot 1.7076^n$ connected vertex covers an these sets can be enumerated in time $O^*(1.7076^n)$ (Wingsternes, 2018).

# Enumeration of connected vertex covers

**Problem (Enumeration of Minimal CVC)**

**Input:** *A connected graph $G$.*

**Task:** *Enumerate all minimal connected vertex covers.*

An $n$-vertex graph has at most $2 \cdot 1.7076^n$ connected vertex covers an these sets can be enumerated in time $O^*(1.7076^n)$ (Wingsternes, 2018).

There are graphs with at least $1.5197^n$ minimal connected vertex covers (Ryland, 2018).

# Enumeration of irredundant sets

A set of vertices $D$ of a graph $G$ is a *irredundant* set if for every $v \in D$ there is a vertex $u \in N[v]$ such that $u$ is not adjacent to other vertices of $D$.

# Enumeration of irredundant sets

A set of vertices $D$ of a graph $G$ is a *irredundant* set if for every $v \in D$ there is a vertex $u \in N[v]$ such that $u$ is not adjacent to other vertices of $D$.

It is said that $u$ is a *private* vertex for $v$.

# Enumeration of irredundant sets

A set of vertices $D$ of a graph $G$ is a *irredundant* set if for every $v \in D$ there is a vertex $u \in N[v]$ such that $u$ is not adjacent to other vertices of $D$.

It is said that $u$ is a *private* vertex for $v$.

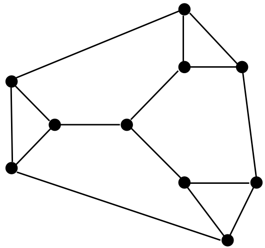
An irredundant set $D$ is maximal if $D$ is an irredundant set and for every $D' \supset D$, $D'$ is not an irredundant set.

# Enumeration of irredundant sets

A set of vertices $D$ of a graph $G$ is a *irredundant* set if for every $v \in D$ there is a vertex $u \in N[v]$ such that $u$ is not adjacent to other vertices of $D$.

It is said that $u$ is a *private* vertex for $v$.

An irredundant set $D$ is maximal if $D$ is an irredundant set and for every $D' \supset D$, $D'$ is not an irredundant set.

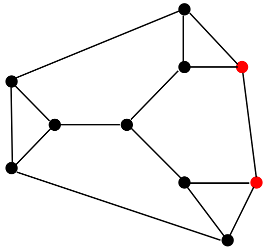


# Enumeration of irredundant sets

A set of vertices $D$ of a graph $G$ is a *irredundant* set if for every $v \in D$ there is a vertex $u \in N[v]$ such that $u$ is not adjacent to other vertices of $D$.

It is said that $u$ is a *private* vertex for $v$.

An irredundant set $D$ is *maximal* if $D$ is an irredundant set and for every $D' \supset D$, $D'$ is not an irredundant set.

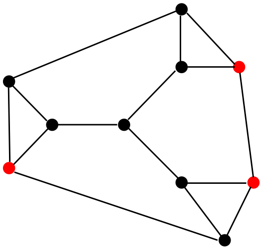


# Enumeration of irredundant sets

A set of vertices $D$ of a graph $G$ is a *irredundant* set if for every $v \in D$ there is a vertex $u \in N[v]$ such that $u$ is not adjacent to other vertices of $D$.

It is said that $u$ is a *private* vertex for $v$.

An irredundant set $D$ is *maximal* if $D$ is an irredundant set and for every $D' \supset D$, $D'$ is not an irredundant set.



# Enumeration of irredundant sets

A set of vertices $D$ of a graph $G$ is a *irredundant* set if for every $v \in D$ there is a vertex $u \in N[v]$ such that $u$ is not adjacent to other vertices of $D$.

It is said that $u$ is a private vertex for $v$.

An irredundant set $D$ is *maximal* if $D$ is an irredundant set and for every $D' \supset D$, $D'$ is not an irredundant set.

# Enumeration of irredundant sets

A set of vertices $D$ of a graph $G$ is a *irredundant* set if for every $v \in D$ there is a vertex $u \in N[v]$ such that $u$ is not adjacent to other vertices of $D$.

It is said that $u$ is a private vertex for $v$.

An irredundant set $D$ is *maximal* if $D$ is an irredundant set and for every $D' \supset D$, $D'$ is not an irredundant set.

Every minimal dominating set is a maximal irredundant set but not the other way around.

# Enumeration of irredundant sets

**Problem (Enumeration of Maximal IS)**

**Input:** *A graph $G$.*

**Task:** *Enumerate all maximal irredundant sets.*

# Enumeration of irredundant sets

**Problem (Enumeration of Maximal IS)**

**Input:** *A graph $G$.*

**Task:** *Enumerate all maximal irredundant sets.*

All maximal irredundant sets of an $n$ vertex graph can be enumerated in time $O^*(2^n)$.

# Enumeration of irredundant sets

## Problem (Enumeration of Maximal IS)

**Input:** *A graph G.*

**Task:** *Enumerate all maximal irredundant sets.*

All maximal irredundant sets of an $n$ vertex graph can be enumerated in time $O^*(2^n)$.

There are graphs with at least $10^{n/5}$ maximal irredundant sets: