HAL
open science

# Enumerating the edge-colourings and total colourings of a regular graph

Stéphane Bessy, Frédéric Havet

## To cite this version:

Stéphane Bessy, Frédéric Havet. Enumerating the edge-colourings and total colourings of a regular graph. 2012 Workshop on Graph Theory and Combinatorics, Aug 2012, National Sun Yat-sen University, Kaohsiung, Taiwan. lirmm-00811571

**HAL Id: lirmm-00811571**

**https://hal-lirmm.ccsd.cnrs.fr/lirmm-00811571**

Submitted on 10 Apr 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Enumerating the edge-colourings of a regular graph

Stéphane Bessy and Frédéric Havet

Assistant Professor
LIRMM, UM2, Montpellier

Researcher
CNRS, Sophia-Antipolis

FRANCE

2012 Workshop on Graph Theory and Combinatorics
Deparment of Applied Mathematics, National Sun Yat-sen University,
Kaohsiung, Taiwan.

# Outline


1. Introduction
   - Colorings
   - Algorithmic problems
   - Our results
2. Enumerating the 3-edge colorings of a cubic graph
   - The 3-edge colorings of a 3-regular graph
   - Turning the proof into algorithm
3. Extensions: $k$-edge colorings and the total colorings
   - $k$-edge colorings
   - A more precise bounds for the 3-edge coloring
   - Total coloring
4. Conclusion

# Introduction


1. Introduction
   - Colorings
   - Algorithmic problems
   - Our results
2. Enumerating the 3-edge colorings of a cubic graph
   - The 3-edge colorings of a 3-regular graph
   - Turning the proof into algorithm
3. Extensions: $k$-edge colorings and the total colorings
   - $k$-edge colorings
   - A more precise bounds for the 3-edge coloring
   - Total coloring
4. Conclusion

# Vertex Coloring

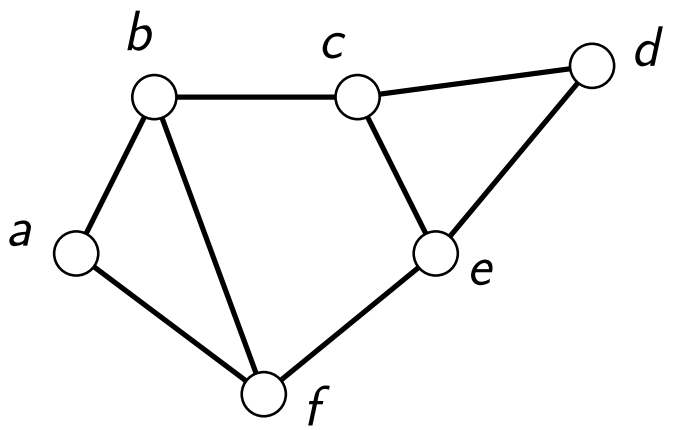
**Definition ($k$-vertex coloring)**

A *$k$-vertex coloring* of a (non oriented) graph $G = (V, E)$ is a function $c : V \to \{1, \ldots, k\}$ such that $uv \in E$ implies $c(u) \neq c(v)$.

# Vertex Coloring

## Definition ($k$-vertex coloring)

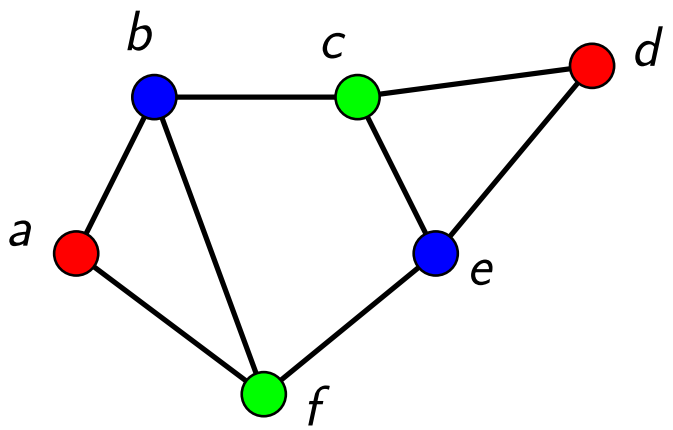
A *$k$-vertex coloring* of a (non oriented) graph $G = (V, E)$ is a function $c : V \to \{1, \ldots, k\}$ such that $uv \in E$ implies $c(u) \neq c(v)$.



# Vertex Coloring

## Definition ($k$-vertex coloring)

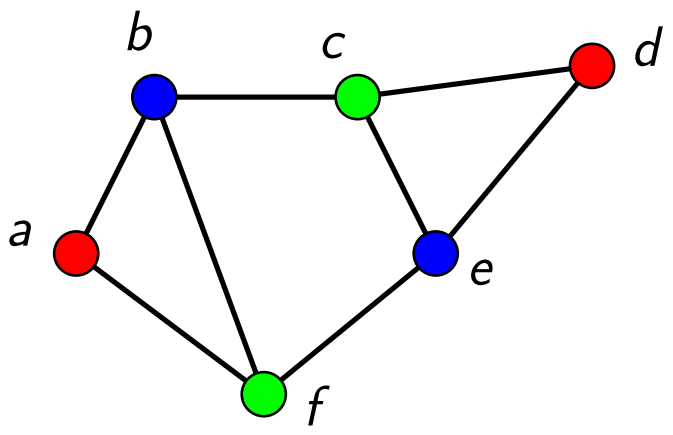
A *$k$-vertex coloring* of a (non oriented) graph $G = (V, E)$ is a function $c : V \to \{1, \ldots, k\}$ such that $uv \in E$ implies $c(u) \neq c(v)$.



# Vertex Coloring

## Definition ($k$-vertex coloring)

A *$k$-vertex coloring* of a (non oriented) graph $G = (V, E)$ is a function $c : V \to \{1, \ldots, k\}$ such that $uv \in E$ implies $c(u) \neq c(v)$.




## Definition (chromatic number)

$\chi(G)$ is the minimum $k$ such that $G$ admits a $k$-vertex coloring.

# Edge Coloring

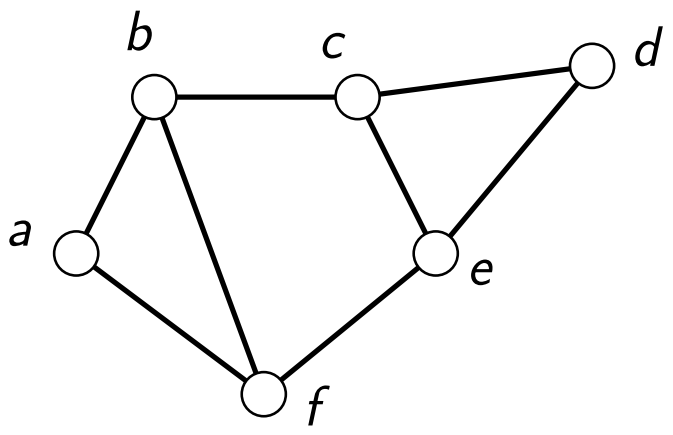
**Definition ($k$-edge coloring)**

A *$k$-edge coloring* of a graph $G = (V, E)$ is a function $c : E \to \{1, \ldots, k\}$ such that $uv \in E$ and $uw \in E$, with $v \neq w$ implies $c(uv) \neq c(uw)$.

# Edge Coloring

## Definition ($k$-edge coloring)

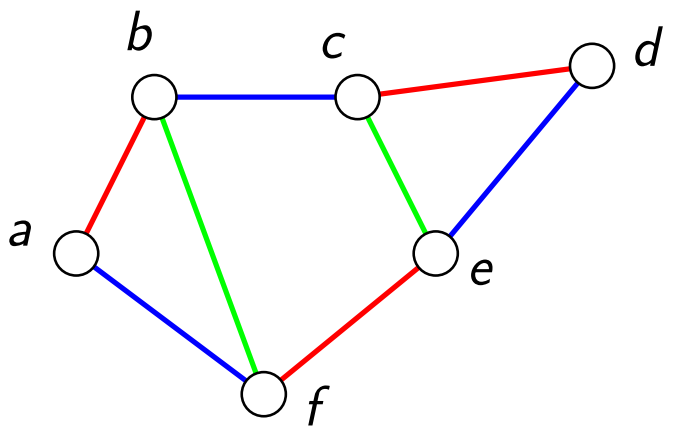
A *$k$-edge coloring* of a graph $G = (V, E)$ is a function $c : E \to \{1, \ldots, k\}$ such that $uv \in E$ and $uw \in E$, with $v \neq w$ implies $c(uv) \neq c(uw)$.



# Edge Coloring

**Definition ($k$-edge coloring)**

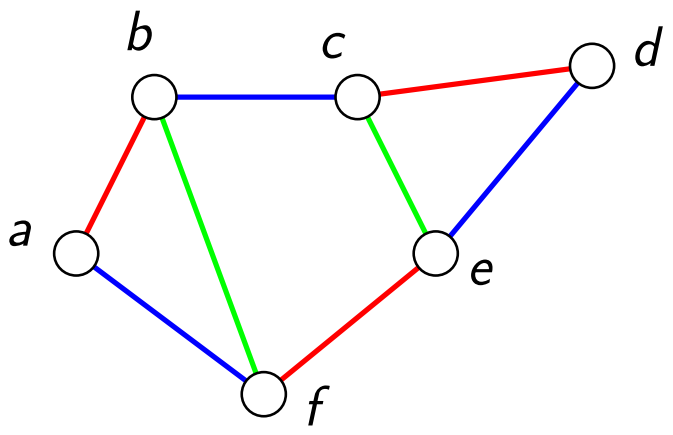
A *$k$-edge coloring* of a graph $G = (V, E)$ is a function $c : E \to \{1, \ldots, k\}$ such that $uv \in E$ and $uw \in E$, with $v \neq w$ implies $c(uv) \neq c(uw)$.



# Edge Coloring

## Definition ($k$-edge coloring)

A *$k$-edge coloring* of a graph $G = (V, E)$ is a function $c : E \to \{1, \ldots, k\}$ such that $uv \in E$ and $uw \in E$, with $v \neq w$ implies $c(uv) \neq c(uw)$.

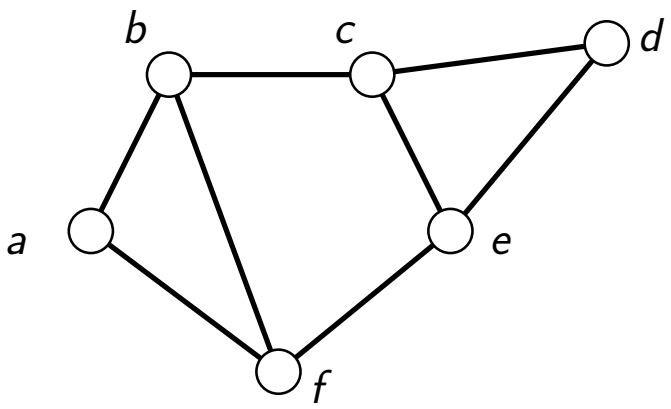



## Definition (chromatic index)

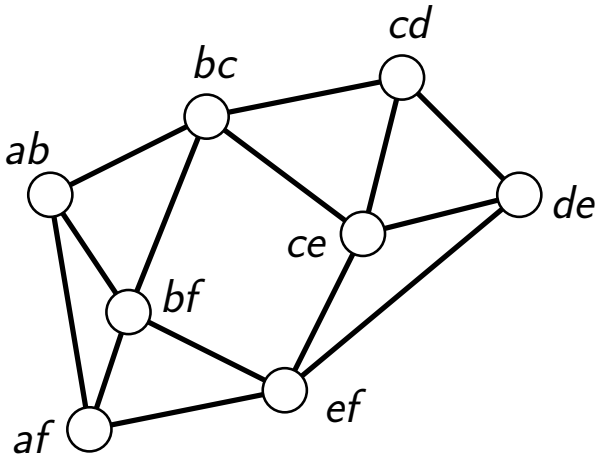
$\chi'(G)$ is the minimum $k$ such that $G$ admits a $k$-edge coloring.

# Edge Coloring

- The *line graph* $L(G)$ of $G$:
  - The vertex set of $L(G)$ is the edge set of $G$
  - Two edges $e$ and $f$ of $G$ are adjacent in $L(G)$ if there are incident in $G$



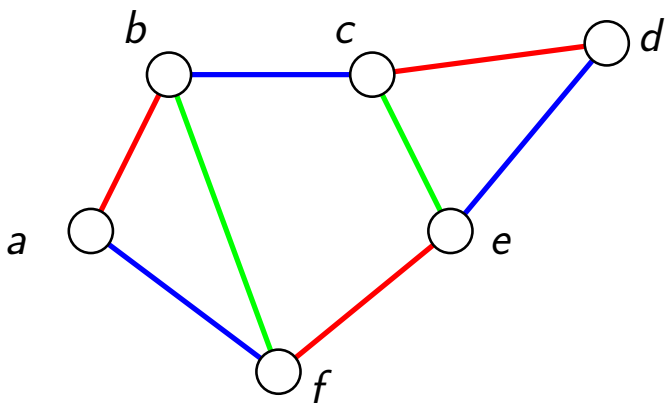

$G$



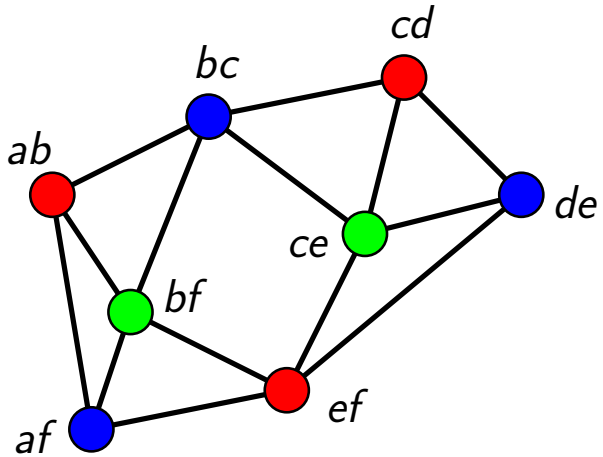

$L(G)$

# Edge Coloring

- The *line graph* $L(G)$ of $G$:
  - The vertex set of $L(G)$ is the edge set of $G$
  - Two edges $e$ and $f$ of $G$ are adjacent in $L(G)$ if there are incident in $G$




$G$




$L(G)$

- $\chi'(G) = \chi(L(G))$

# Total Coloring

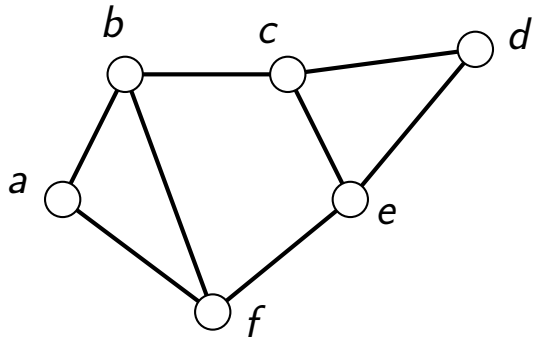
## Definition ($k$-total coloring)

A *$k$-total coloring* of $G = (V, E)$ is $c : V \cup E \to \{1, \ldots, k\}$ such that:
- $uv \in E$ implies $c(u) \neq c(v)$
- $uv \in E$ and $uw \in E$, with $v \neq w$ implies $c(uv) \neq c(uw)$
- $uv \in E$ implies $c(uv) \neq c(u)$

# Total Coloring

## Definition ($k$-total coloring)

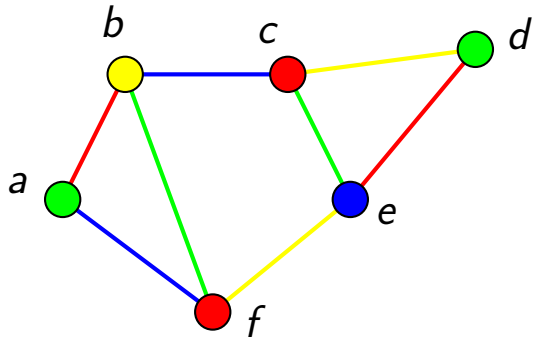
A *$k$-total coloring* of $G = (V, E)$ is $c : V \cup E \to \{1, \ldots, k\}$ such that:
- $uv \in E$ implies $c(u) \neq c(v)$
- $uv \in E$ and $uw \in E$, with $v \neq w$ implies $c(uv) \neq c(uw)$
- $uv \in E$ implies $c(uv) \neq c(u)$



# Total Coloring

## Definition ($k$-total coloring)

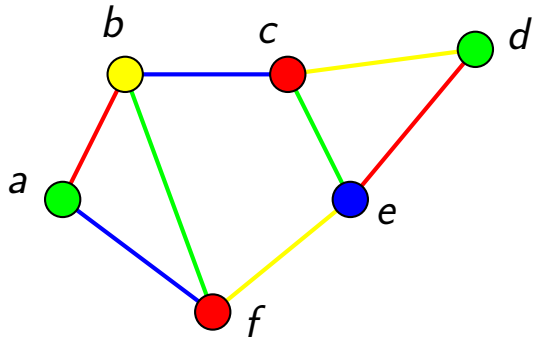
A *$k$-total coloring* of $G = (V, E)$ is $c : V \cup E \to \{1, \ldots, k\}$ such that:
- $uv \in E$ implies $c(u) \neq c(v)$
- $uv \in E$ and $uw \in E$, with $v \neq w$ implies $c(uv) \neq c(uw)$
- $uv \in E$ implies $c(uv) \neq c(u)$



# Total Coloring

## Definition ($k$-total coloring)

A *$k$-total coloring* of $G = (V, E)$ is $c : V \cup E \to \{1, \ldots, k\}$ such that:
- $uv \in E$ implies $c(u) \neq c(v)$
- $uv \in E$ and $uw \in E$, with $v \neq w$ implies $c(uv) \neq c(uw)$
- $uv \in E$ implies $c(uv) \neq c(u)$




## Definition (total chromatic number)

$\chi_T(G)$ is the minimum $k$ such that $G$ admits a $k$-total coloring.

# Famous bounds

Let $\Delta(G)$ denotes the maximum degree of $G$.

# Famous bounds

Let $\Delta(G)$ denotes the maximum degree of $G$.

- Brook's Theorem (1941): $\chi(G) \leq \Delta(G)$ if $G$ is not a complete graph or an odd cycle.

# Famous bounds

Let $\Delta(G)$ denotes the maximum degree of $G$.

- Brook's Theorem (1941): $\chi(G) \leq \Delta(G)$ if $G$ is not a complete graph or an odd cycle.
- Vizing's Theorem (1964): $\Delta(G) \leq \chi'(G) \leq \Delta(G) + 1$

# Famous bounds

Let $\Delta(G)$ denotes the maximum degree of $G$.

- Brook's Theorem (1941): $\chi(G) \leq \Delta(G)$ if $G$ is not a complete graph or an odd cycle.
- Vizing's Theorem (1964): $\Delta(G) \leq \chi'(G) \leq \Delta(G) + 1$
- Total coloring conjecture (Behzad, Vizing, ∼1964): $\Delta(G) + 1 \leq \chi_T(G) \leq \Delta(G) + 2$

# Algorithmic problems

$k$-VERTEX/EDGE/TOTAL COLORING:

- *Input:* a graph $G = (V, E)$
- *Output:* Does $G$ admit a $k$-vertex/edge/total coloring?

# Algorithmic problems

$k$-VERTEX/EDGE/TOTAL COLORING:

- *Input:* a graph $G = (V, E)$
- *Output:* Does $G$ admit a $k$-vertex/edge/total coloring?

- $k$-VERTEX COLORING is NP-complete for every fixed $k \geq 3$ (M.R. Garey and D.S. Johnson, 1979).

# Algorithmic problems

$k$-VERTEX/EDGE/TOTAL COLORING:

- *Input:* a graph $G = (V, E)$
- *Output:* Does $G$ admit a $k$-vertex/edge/total coloring?

- $k$-VERTEX COLORING is NP-complete for every fixed $k \geq 3$ (M.R. Garey and D.S. Johnson, 1979).
- $k$-EDGE COLORING is NP-complete for every fixed $k \geq 3$ (I. Holyer, 1981).

# Algorithmic problems

$k$-VERTEX/EDGE/TOTAL COLORING:

- *Input:* a graph $G = (V, E)$
- *Output:* Does $G$ admit a $k$-vertex/edge/total coloring?

- $k$-VERTEX COLORING is NP-complete for every fixed $k \geq 3$ (M.R. Garey and D.S. Johnson, 1979).
- $k$-EDGE COLORING is NP-complete for every fixed $k \geq 3$ (I. Holyer, 1981).
- $k$-TOTAL COLORING is NP-complete for every fixed $k \geq 4$ (A. Sánchez-Arroyo, 1989).

# Algorithmic problems

$k$-VERTEX/EDGE/TOTAL COLORING:

- *Input:* a graph $G = (V, E)$
- *Output:* Does $G$ admit a $k$-vertex/edge/total coloring?

- $k$-VERTEX COLORING is NP-complete for every fixed $k \geq 3$ (M.R. Garey and D.S. Johnson, 1979).
- $k$-EDGE COLORING is NP-complete for every fixed $k \geq 3$ (I. Holyer, 1981).
- $k$-TOTAL COLORING is NP-complete for every fixed $k \geq 4$ (A. Sánchez-Arroyo, 1989).

ENUM-$k$-VERTEX/EDGE/TOTAL COLORING:

- *Input:* a graph $G = (V, E)$
- *Output:* Enumerate all the $k$-vertex/edge/total coloring of $G$.

# Algorithmic problems

$k$-VERTEX/EDGE/TOTAL COLORING:

- *Input:* a graph $G = (V, E)$
- *Output:* Does $G$ admit a $k$-vertex/edge/total coloring?

- $k$-VERTEX COLORING is NP-complete for every fixed $k \geq 3$ (M.R. Garey and D.S. Johnson, 1979).
- $k$-EDGE COLORING is NP-complete for every fixed $k \geq 3$ (I. Holyer, 1981).
- $k$-TOTAL COLORING is NP-complete for every fixed $k \geq 4$ (A. Sánchez-Arroyo, 1989).

ENUM-$k$-VERTEX/EDGE/TOTAL COLORING:

- *Input:* a graph $G = (V, E)$
- *Output:* Enumerate all the $k$-vertex/edge/total coloring of $G$.

# Algorithmic point of view

What to do with an NP-complete problem?

# Algorithmic point of view

What to do with an NP-complete problem?

- *approximation algorithm*: find in poly-time a result which is not worst than $c$ time the optimal.
- *randomized algorithm*: find a result which is good w.h.p. or/and with running time correct w.h.p.
- ...

# Algorithmic point of view

What to do with an NP-complete problem?

- *approximation algorithm*: find in poly-time a result which is not worst than $c$ time the optimal.
- *randomized algorithm*: find a result which is good w.h.p. or/and with running time correct w.h.p.
- ...
- *parameterized algorithm*: find an algorithm with running time $f(k).poly(n)$ for some parameter $k$.

# Algorithmic point of view

What to do with an NP-complete problem?

- *approximation algorithm*: find in poly-time a result which is not worst than $c$ time the optimal.
- *randomized algorithm*: find a result which is good w.h.p. or/and with running time correct w.h.p.
- ...
- *parameterized algorithm*: find an algorithm with running time $f(k).poly(n)$ for some parameter $k$.
- **exact (exponential) algorithms**: find an algorithm with running time $O^\star(c^n)$ $(= O(P(n)c^n))$

# Examples of exact algorithms

ENUM-3-EDGE COLORING:

- *Input:* a graph $G = (V, E)$ with $\Delta(G) \leq 3$.
- *Output:* Enumerate all the 3-vertex/edge/total coloring of $G$.

# Examples of exact algorithms

ENUM-3-EDGE COLORING:

- *Input:* a graph $G = (V, E)$ with $\Delta(G) \leq 3$.
- *Output:* Enumerate all the 3-vertex/edge/total coloring of $G$.

The exhaustive algorithm:

- Try every possible colorings of the edge set and decide which colorings are proper.

# Examples of exact algorithms

ENUM-3-EDGE COLORING:

- *Input:* a graph $G = (V, E)$ with $\Delta(G) \leq 3$.
- *Output:* Enumerate all the 3-vertex/edge/total coloring of $G$.

The exhaustive algorithm:

- Try every possible colorings of the edge set and decide which colorings are proper.

Time analysis:
$O^\star(3^m) = O^\star(3^{\frac{3}{2}n}) = O^\star(5.1962^n)$

# Examples of exact algorithms

ENUM-3-EDGE COLORING:

- *Input:* a graph $G = (V, E)$ with $\Delta(G) \leq 3$.
- *Output:* Enumerate all the 3-vertex/edge/total coloring of $G$.

The exhaustive algorithm:

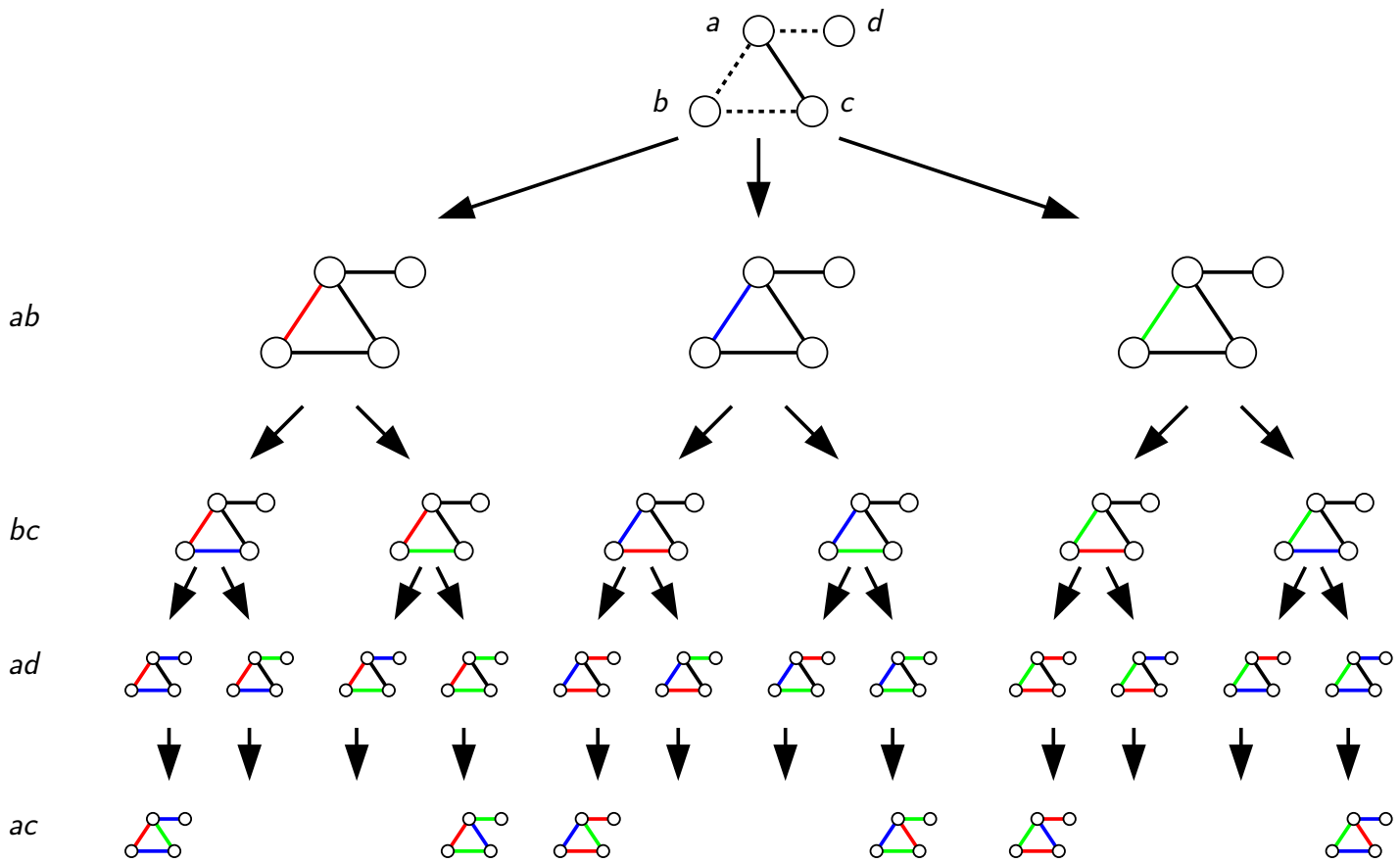
- Try every possible colorings of the edge set and decide which colorings are proper.

Time analysis:
$O^\star(3^m) = O^\star(3^{\frac{3}{2}n}) = O^\star(5.1962^n)$

Usual methods to improve: **branching algorithms** and dynamic programming.

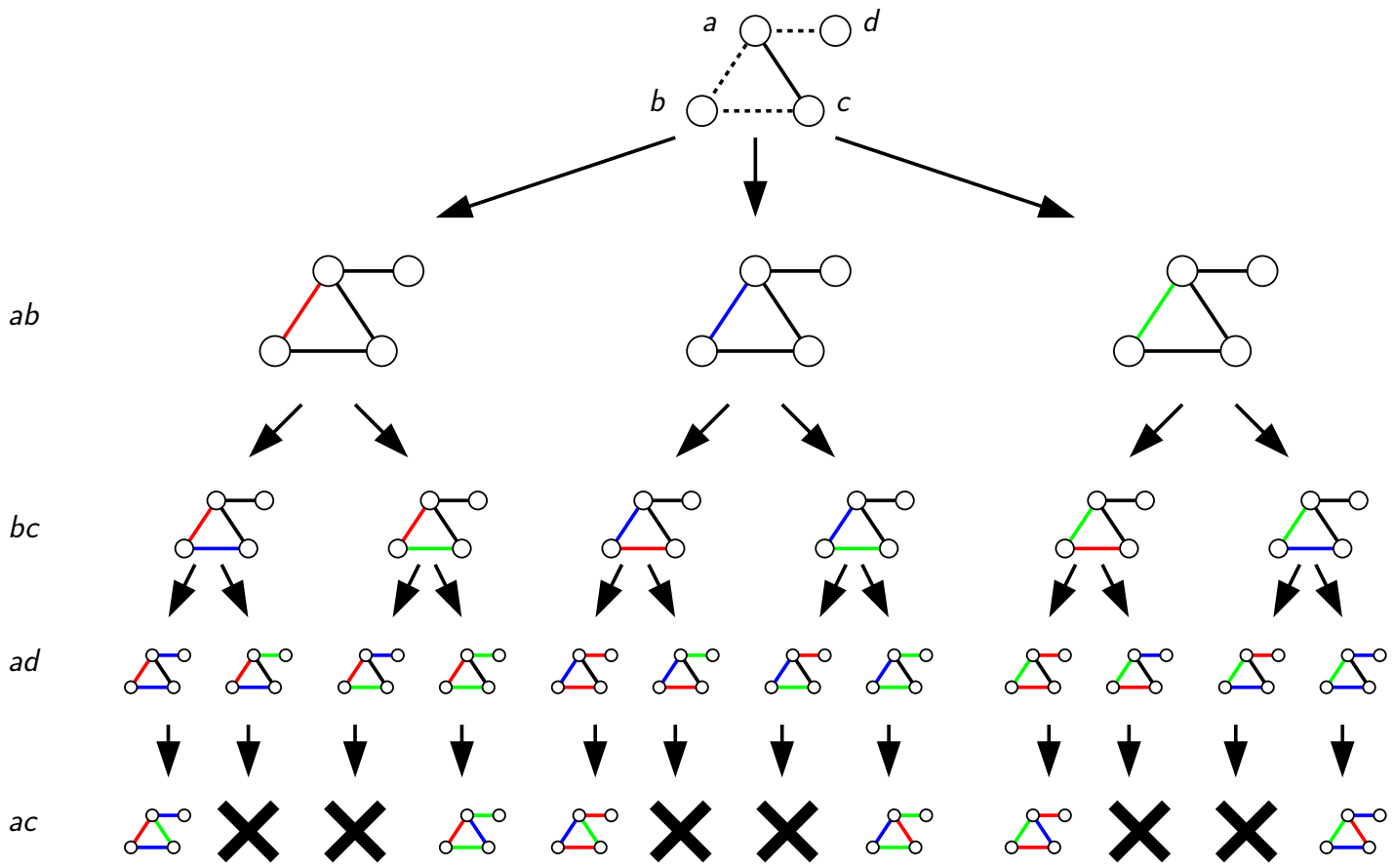
# Examples of exact algorithms

ENUM-3-EDGE COLORING: a 'naive' branching algorithms



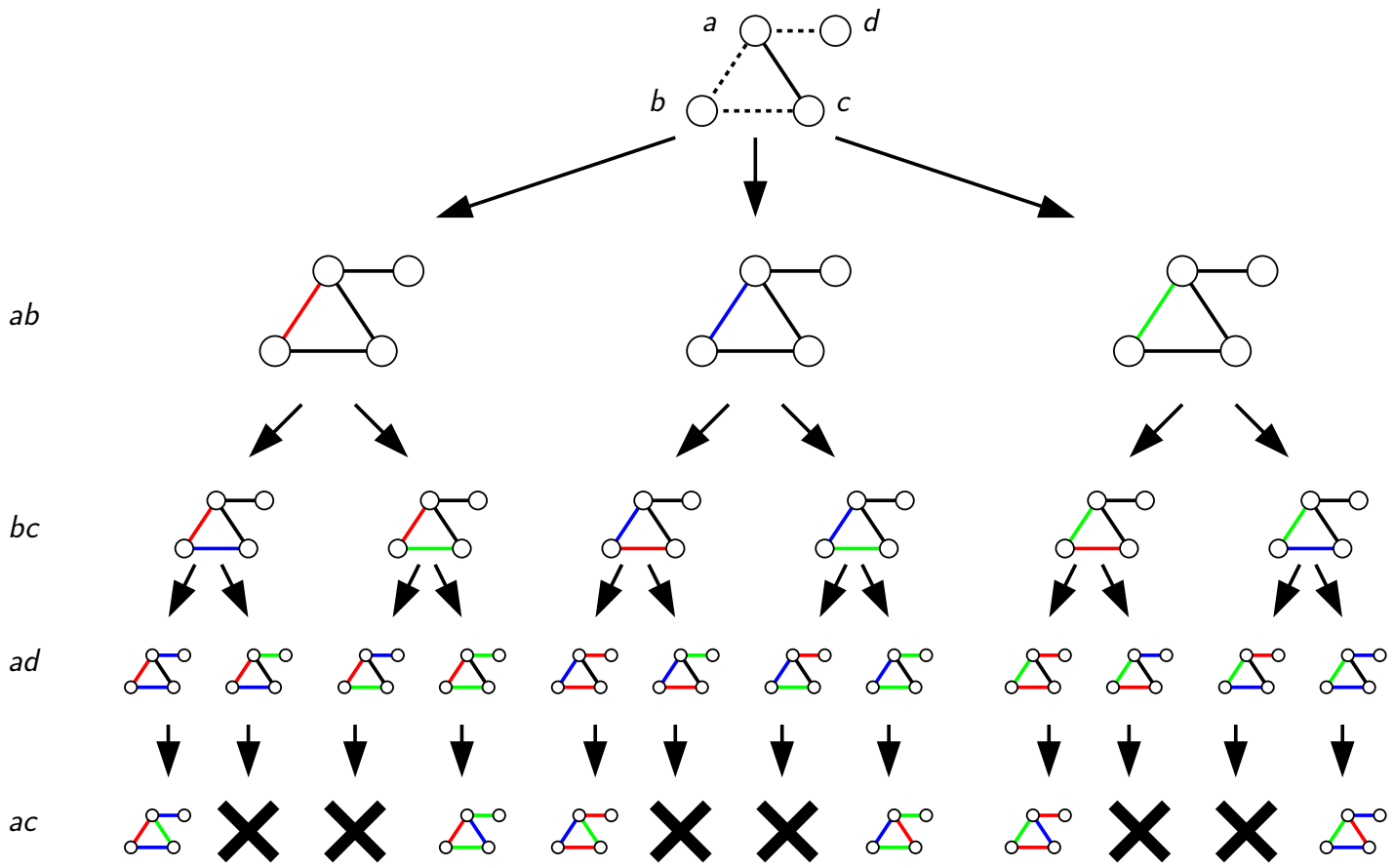
# Examples of exact algorithms

ENUM-3-EDGE COLORING: a 'naive' branching algorithms



# Examples of exact algorithms

ENUM-3-EDGE COLORING: a 'naive' branching algorithms

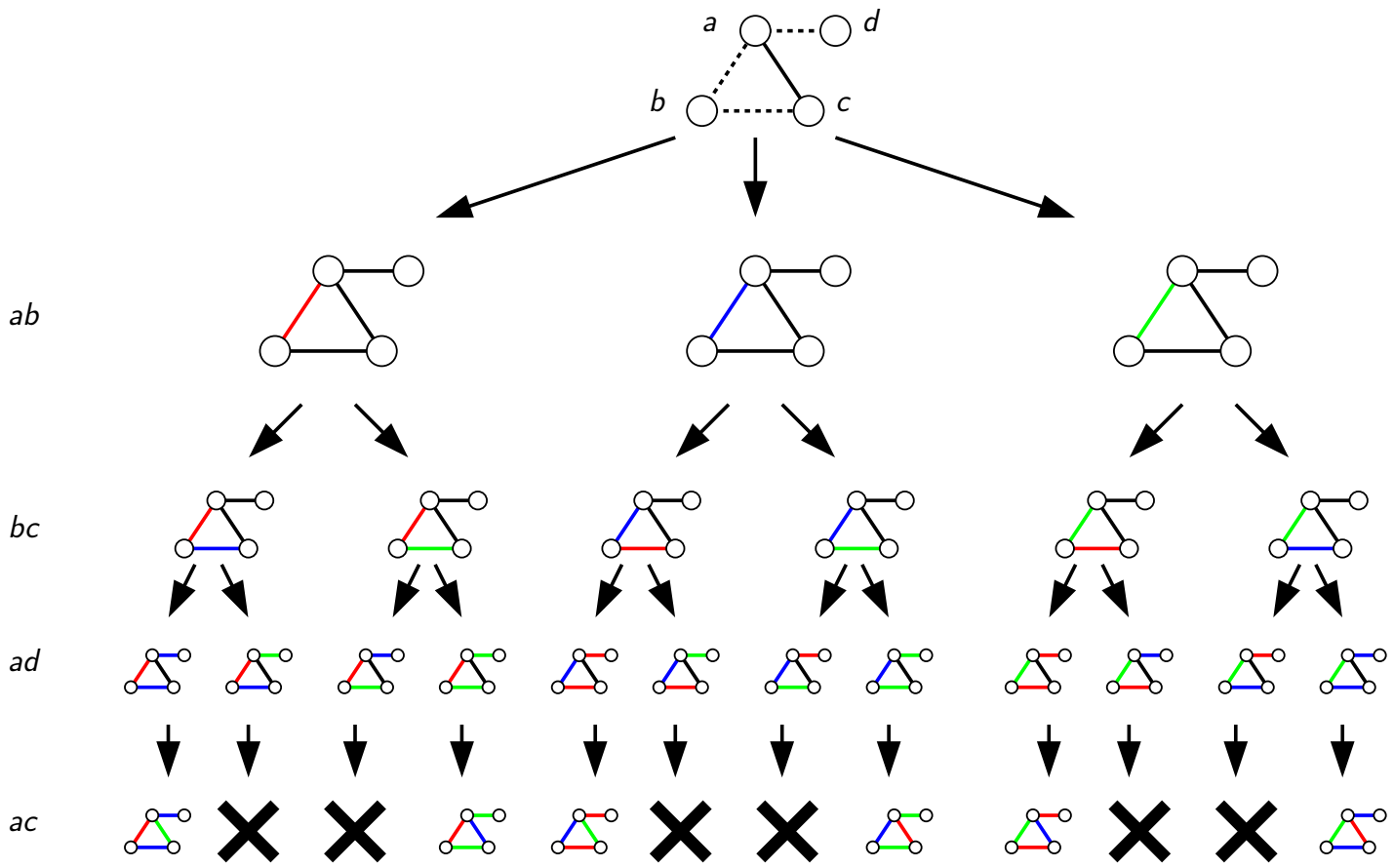



Time analysis: $O^\star(2^m) = O^\star(2^{\frac{3}{2}n}) = O^\star(2.8284^n)$

Polynomial space.

# Examples of exact algorithms

ENUM-3-EDGE COLORING: a 'naive' branching algorithms




Time analysis: $O^\star(2^m) = O^\star(2^{\frac{3}{2}n}) = O^\star(2.8284^n)$ min. nb of leaves

Polynomial space. min. height

# Previous work: exact algorithms

What is the gain on the running time?

# Previous work: exact algorithms

What is the gain on the running time?

In the same environment:
$t = O^\star(a^{n_1}) = O^\star(b^{n_2})$ with $a < b$

We obtain: $n_1 \sim \frac{\log b}{\log a} n_2$
A multiplicative factor on the instance size of $\frac{\log b}{\log a} > 1$

# Previous work: exact algorithms

What is the gain on the running time?

In the same environment:
$t = O^\star(a^{n_1}) = O^\star(b^{n_2})$ with $a < b$

We obtain: $n_1 \sim \frac{\log b}{\log a} n_2$
A multiplicative factor on the instance size of $\frac{\log b}{\log a} > 1$

Here: $\frac{\log 3^{\frac{3}{2}}}{\log 2^{\frac{3}{2}}} = 1.58$

# Previous work: exact algorithms

Main results in exact algorithms to color a graph:

- 3-VERTEX COLORING in $O^\star(1.3289^n)$ (R. Beigel and D. Eppstein, 2005)

# Previous work: exact algorithms

Main results in exact algorithms to color a graph:

- 3-VERTEX COLORING in $O^\star(1.3289^n)$ (R. Beigel and D. Eppstein, 2005)
- MINIMUM VERTEX COLORING in $O^\star(2^n)$ (A. Bjöklund, T. Husfeld and M. Koivisto, 2006)

# Previous work: exact algorithms

Main results in exact algorithms to color a graph:

- 3-VERTEX COLORING in $O^\star(1.3289^n)$ (R. Beigel and D. Eppstein, 2005)
- MINIMUM VERTEX COLORING in $O^\star(2^n)$ (A. Bjöklund, T. Husfeld and M. Koivisto, 2006)
- ENUM 3-VERTEX COLORING in $O^\star(1.6259^n)$ (F. Fomin, S. Gasper and S. Saurabh, 2007)

# Previous work: exact algorithms

Main results in exact algorithms to color a graph:

- 3-VERTEX COLORING in $O^\star(1.3289^n)$ (R. Beigel and D. Eppstein, 2005)
- MINIMUM VERTEX COLORING in $O^\star(2^n)$ (A. Bjöklund, T. Husfeld and M. Koivisto, 2006)
- ENUM 3-VERTEX COLORING in $O^\star(1.6259^n)$ (F. Fomin, S. Gasper and S. Saurabh, 2007)
- 3-EDGE COLORING in $O^\star(1.344^n)$ (L. Kowalik, 2009)

# Previous work: exact algorithms

Main results in exact algorithms to color a graph:

- 3-VERTEX COLORING in $O^\star(1.3289^n)$ (R. Beigel and D. Eppstein, 2005)
- MINIMUM VERTEX COLORING in $O^\star(2^n)$ (A. Bjöklund, T. Husfeld and M. Koivisto, 2006)
- ENUM 3-VERTEX COLORING in $O^\star(1.6259^n)$ (F. Fomin, S. Gasper and S. Saurabh, 2007)
- 3-EDGE COLORING in $O^\star(1.344^n)$ (L. Kowalik, 2009)
- ENUM-3-EDGE COLORING in $O^\star(1.5423^n)$ (P.A. Golovatch, D. Kratsch and J.F. Couturier, 2010)

# Previous work: exact algorithms

Main results in exact algorithms to color a graph:

- 3-VERTEX COLORING in $O^\star(1.3289^n)$ (R. Beigel and D. Eppstein, 2005)
- MINIMUM VERTEX COLORING in $O^\star(2^n)$ (A. Bjöklund, T. Husfeld and M. Koivisto, 2006)
- ENUM 3-VERTEX COLORING in $O^\star(1.6259^n)$ (F. Fomin, S. Gasper and S. Saurabh, 2007)
- 3-EDGE COLORING in $O^\star(1.344^n)$ (L. Kowalik, 2009)
- ENUM-3-EDGE COLORING in $O^\star(1.5423^n)$ (P.A. Golovatch, D. Kratsch and J.F. Couturier, 2010)
- ENUM-4-TOTAL COLORING in $O^\star(3.0845^n)$ (P.A. Golovatch, D. Kratsch and J.F. Couturier, 2010)

# Previous work: exact algorithms

Main results in exact algorithms to color a graph:

- 3-VERTEX COLORING in $O^\star(1.3289^n)$ (R. Beigel and D. Eppstein, 2005)
- MINIMUM VERTEX COLORING in $O^\star(2^n)$ (A. Bjöklund, T. Husfeld and M. Koivisto, 2006)
- ENUM 3-VERTEX COLORING in $O^\star(1.6259^n)$ (F. Fomin, S. Gasper and S. Saurabh, 2007)
- 3-EDGE COLORING in $O^\star(1.344^n)$ (L. Kowalik, 2009)
- ENUM-3-EDGE COLORING in $O^\star(1.5423^n)$ (P.A. Golovatch, D. Kratsch and J.F. Couturier, 2010)
- ENUM-4-TOTAL COLORING in $O^\star(3.0845^n)$ (P.A. Golovatch, D. Kratsch and J.F. Couturier, 2010)

# Our results

Motivations:

- a useful technique for EDGE COLORING

# Our results

Motivations:

- a useful technique for EDGE COLORING
- 3-EDGE COLORING more interesting

# Our results

Motivations:

- a useful technique for EDGE COLORING
- 3-EDGE COLORING more interesting
- Gap: $\Omega(1.2822^n) \leq$ ENUM-3-EDGE COLORING $\leq O^\star(1.5423^n)$

# Our results

Motivations:

- a useful technique for EDGE COLORING
- 3-EDGE COLORING more interesting
- Gap: $\Omega(1.2822^n) \leq$ ENUM-3-EDGE COLORING $\leq O^\star(1.5423^n)$
- ? $\leq$ ENUM-4-TOTAL COLORING $\leq O^\star(3.0845^n)$

# Our results

Motivations:

- a useful technique for EDGE COLORING
- 3-EDGE COLORING more interesting
- Gap: $\Omega(1.2822^n) \leq$ ENUM-3-EDGE COLORING $\leq O^\star(1.5423^n)$
- ? $\leq$ ENUM-4-TOTAL COLORING $\leq O^\star(3.0845^n)$

Our results:

- ENUM-3-EDGE COLORING in $O^\star(1.4142^n)$ and a sharp example (multi-graph). (factor on data size: 1.25)

# Our results

Motivations:

- a useful technique for EDGE COLORING
- 3-EDGE COLORING more interesting
- Gap: $\Omega(1.2822^n) \leq$ ENUM-3-EDGE COLORING $\leq O^\star(1.5423^n)$
- ? $\leq$ ENUM-4-TOTAL COLORING $\leq O^\star(3.0845^n)$

Our results:

- ENUM-3-EDGE COLORING in $O^\star(1.4142^n)$ and a sharp example (multi-graph). (factor on data size: 1.25)
- ENUM-4-TOTAL COLORING in $O^\star(2.8285^n)$. Example in $\Omega(2.3784^n)$.

# Our results

Motivations:

- a useful technique for EDGE COLORING
- 3-EDGE COLORING more interesting
- Gap: $\Omega(1.2822^n) \leq$ ENUM-3-EDGE COLORING $\leq O^\star(1.5423^n)$
- ? $\leq$ ENUM-4-TOTAL COLORING $\leq O^\star(3.0845^n)$

Our results:

- ENUM-3-EDGE COLORING in $O^\star(1.4142^n)$ and a sharp example (multi-graph). (factor on data size: 1.25)
- ENUM-4-TOTAL COLORING in $O^\star(2.8285^n)$. Example in $\Omega(2.3784^n)$.
- ENUM-$k$-EDGE COLORING in $O^\star((k-1!)^{\frac{n}{2}})$ and a sharp example (multi-graph).

# Our results

Motivations:

- a useful technique for EDGE COLORING
- 3-EDGE COLORING more interesting
- Gap: $\Omega(1.2822^n) \leq$ ENUM-3-EDGE COLORING $\leq O^\star(1.5423^n)$
- ? $\leq$ ENUM-4-TOTAL COLORING $\leq O^\star(3.0845^n)$

Our results:

- ENUM-3-EDGE COLORING in $O^\star(1.4142^n)$ and a sharp example (multi-graph). (factor on data size: 1.25)
- ENUM-4-TOTAL COLORING in $O^\star(2.8285^n)$. Example in $\Omega(2.3784^n)$.
- ENUM-$k$-EDGE COLORING in $O^\star((k-1!)^{\frac{n}{2}})$ and a sharp example (multi-graph).
- More precise bound on the number of 3-edge coloring of simple graph

# Enumerating the 3-edge colorings


1. Introduction
   - Colorings
   - Algorithmic problems
   - Our results
2. Enumerating the 3-edge colorings of a cubic graph
   - The 3-edge colorings of a 3-regular graph
   - Turning the proof into algorithm
3. Extensions: $k$-edge colorings and the total colorings
   - $k$-edge colorings
   - A more precise bounds for the 3-edge coloring
   - Total coloring
4. Conclusion

# Enum-3-edge coloring

We denote by $c_k(G)$ the number of $k$-edge-colorings of a graph $G$.

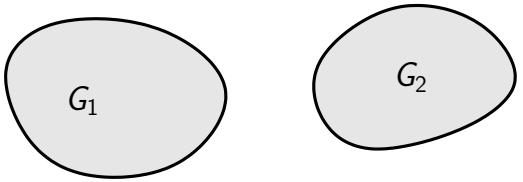
Here, we want to compute $c_3(G)$, $G$ being a sub-cubic (multiple) graph.

# Enum-3-edge coloring

We denote by $c_k(G)$ the number of $k$-edge-colorings of a graph $G$.

Here, we want to compute $c_3(G)$, $G$ being a sub-cubic (multiple) graph.

We can assume that $G$ is connected.



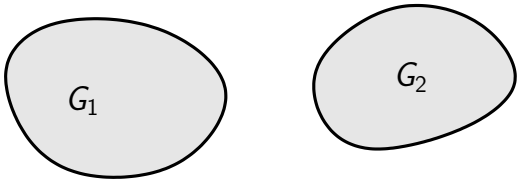

$$c_3(G) = c_3(G_1) \cdot c_3(G_2)$$

# Enum-3-edge coloring

We denote by $c_k(G)$ the number of $k$-edge-colorings of a graph $G$.

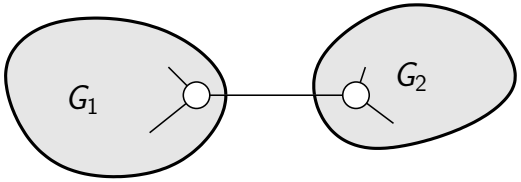
Here, we want to compute $c_3(G)$, $G$ being a sub-cubic (multiple) graph.

We can assume that $G$ is connected.




$$c_3(G) = c_3(G_1) \cdot c_3(G_2)$$

We can assume that $G$ is 2-(vertex) connected.




$$c_3(G) = \frac{1}{3} \cdot c_3(G_1) \cdot c_3(G_2)$$

# Number of 3-edge colorings of 3-regular graphs

**Lemma**

*Let $C_n$ be the cycle of length $n$.*

$$c_3(C_n) = \begin{cases} 2^n + 2, & \text{if } n \text{ is even,} \\ 2^n - 2, & \text{if } n \text{ is odd.} \end{cases}$$

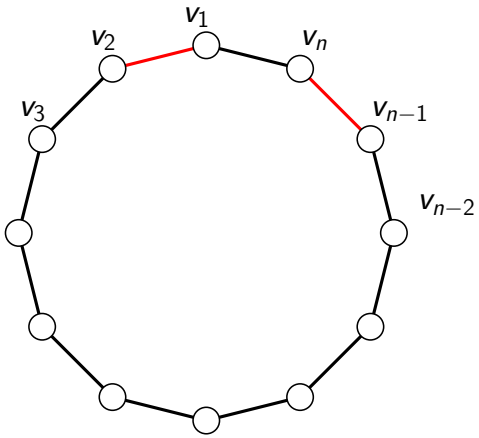
# Number of 3-edge colorings of 3-regular graphs

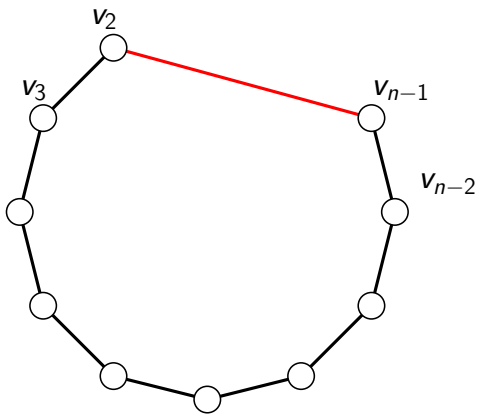
**Lemma**

*Let $C_n$ be the cycle of length $n$.*

$$c_3(C_n) = \begin{cases} 2^n + 2, & \text{if } n \text{ is even,} \\ 2^n - 2, & \text{if } n \text{ is odd.} \end{cases}$$

**Proof:** By induction on $n$. True for $n = 2$ and $n = 3$.




$$c_3(C_n) = 2.c_3(C_{n-2}) +$$

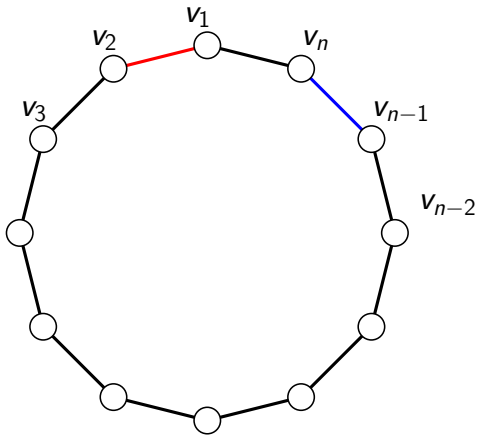
# Number of 3-edge colorings of 3-regular graphs

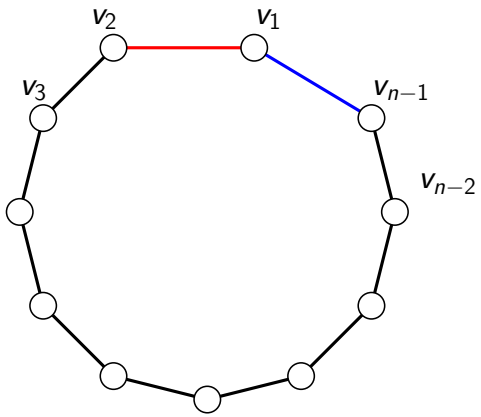
## Lemma

*Let $C_n$ be the cycle of length $n$.*

$$c_3(C_n) = \begin{cases} 2^n + 2, & \text{if } n \text{ is even,} \\ 2^n - 2, & \text{if } n \text{ is odd.} \end{cases}$$

**Proof:** By induction on $n$. True for $n = 2$ and $n = 3$.




$$c_3(C_n) = 2.c_3(C_{n-2}) + c_3(C_{n-1}) \qquad \square$$

# Number of 3-edge colorings of 3-regular graphs

We denote by $n_i$ the number of vertices of degree $i$ in $G$.

**Theorem**

*Let $G$ be a 2-connected subcubic graph. Then $c_3(G) \leq 3 \cdot 2^{n-\frac{n_3}{2}}$.*

# Number of 3-edge colorings of 3-regular graphs

We denote by $n_i$ the number of vertices of degree $i$ in $G$.

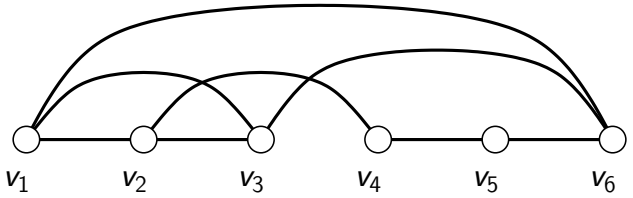
**Theorem**

*Let $G$ be a 2-connected subcubic graph. Then $c_3(G) \leq 3 \cdot 2^{n-\frac{n_3}{2}}$.*

**Proof:** If $G$ is a cycle, it is true.
Otherwise, let $v_1$ and $v_n$ be two vertices of degree 3 and consider $v_1, \ldots, v_n$ an *st-ordering* of $G$:

for all $1 < i < n$, $\quad d(v_i)_{\{v_1,\ldots,v_{i-1}\}} \geq 1$ and $d(v_i)_{\{v_{i+1},\ldots,v_n\}} \geq 1$.

(always exists: Lempel et al.,1967)

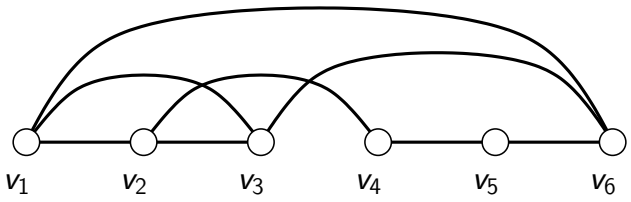


# Number of 3-edge colorings of 3-regular graphs

**Theorem**

*Let $G$ be a 2-connected subcubic graph. Then $c_3(G) \leq 3 \cdot 2^{n-\frac{n_3}{2}}$.*

**Proof:**



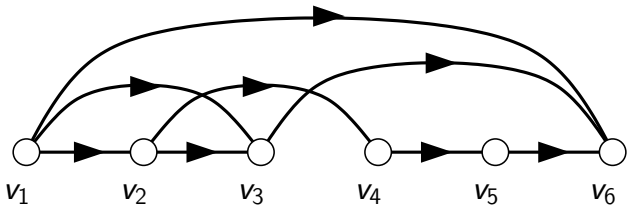

Orient from left to right.
$A^+ = \{v \ : \ d^+(v) = 2 \text{ and } d^-(v) = 1\}$
$A^- = \{v \ : \ d^+(v) = 1 \text{ and } d^-(v) = 2\}$
We have $|A^+| = |A^-| = (n_3 - 2)/2$

# Number of 3-edge colorings of 3-regular graphs

**Theorem**

*Let $G$ be a 2-connected subcubic graph. Then $c_3(G) \leq 3 \cdot 2^{n-\frac{n_3}{2}}$.*

**Proof:**




Orient from left to right.
$A^+ = \{v \ : \ d^+(v) = 2 \text{ and } d^-(v) = 1\}$
$A^- = \{v \ : \ d^+(v) = 1 \text{ and } d^-(v) = 2\}$
We have $|A^+| = |A^-| = (n_3 - 2)/2$

# Number of 3-edge colorings of 3-regular graphs

**Theorem**

*Let $G$ be a 2-connected subcubic graph. Then $c_3(G) \leq 3 \cdot 2^{n-\frac{n_3}{2}}$.*

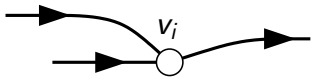
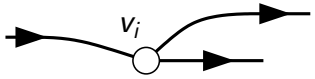
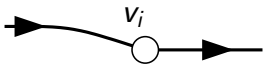
**Proof:** We have $|A^+| = |A^-| = (n_3 - 2)/2$
Denote by $c_i$ the number of partial 3-edge colorings of arcs with tail in $\{v_1, \ldots, v_i\}$.

# Number of 3-edge colorings of 3-regular graphs

**Theorem**

*Let $G$ be a 2-connected subcubic graph. Then $c_3(G) \leq 3 \cdot 2^{n-\frac{n_3}{2}}$.*

**Proof:** We have $|A^+| = |A^-| = (n_3 - 2)/2$
Denote by $c_i$ the number of partial 3-edge colorings of arcs with tail in $\{v_1, \ldots, v_i\}$.

- $c_1 = 6$
- if $v_i \in A^-$  $c_i \leq c_{i-1}$
- if $v_i \in A^+$  $c_i \leq 2c_{i-1}$
- if $d^+(v_i) = d^-(v_i) = 1$  $c_i \leq 2c_{i-1}$

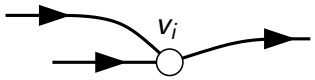
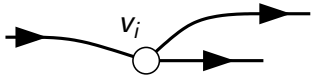
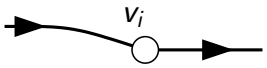
# Number of 3-edge colorings of 3-regular graphs

**Theorem**

*Let $G$ be a 2-connected subcubic graph. Then $c_3(G) \leq 3 \cdot 2^{n-\frac{n_3}{2}}$.*

**Proof:** We have $|A^+| = |A^-| = (n_3 - 2)/2$

Denote by $c_i$ the number of partial 3-edge colorings of arcs with tail in $\{v_1, \ldots, v_i\}$.

- $c_1 = 6$
- if $v_i \in A^-$  $c_i \leq c_{i-1}$
- if $v_i \in A^+$  $c_i \leq 2c_{i-1}$
- if $d^+(v_i) = d^-(v_i) = 1$  $c_i \leq 2c_{i-1}$

$c_3(G) \leq 6 \cdot 2^{(n-2)-(n_3-2)/2} = 3 \cdot 2^{n-\frac{n_3}{2}}$ □

# Number of 3-edge colorings of 3-regular graphs

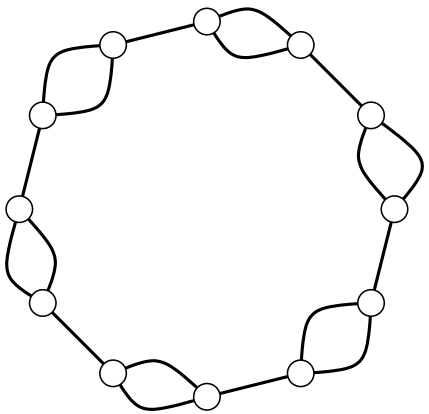
**Theorem**

*Let $G$ be a 2-connected subcubic graph. Then $c_3(G) \leq 3 \cdot 2^{n-\frac{n_3}{2}}$.*

**Corollary**

*Let $G$ be a cubic graph, then $c_3(G) \leq 3 \cdot 2^{\frac{n}{2}}$.*

This is sharp for multi-graphs:



# Enum-3-edge coloring:

**Corollary (Solving Enum-3-edge coloring:)**

*There exists a branching algorithm with running time* $O^\star(2^{\frac{n}{2}}) = O^\star(1.4143^n)$ *and polynomial space to solve* Enum-3-edge coloring.

# Enum-3-edge coloring:

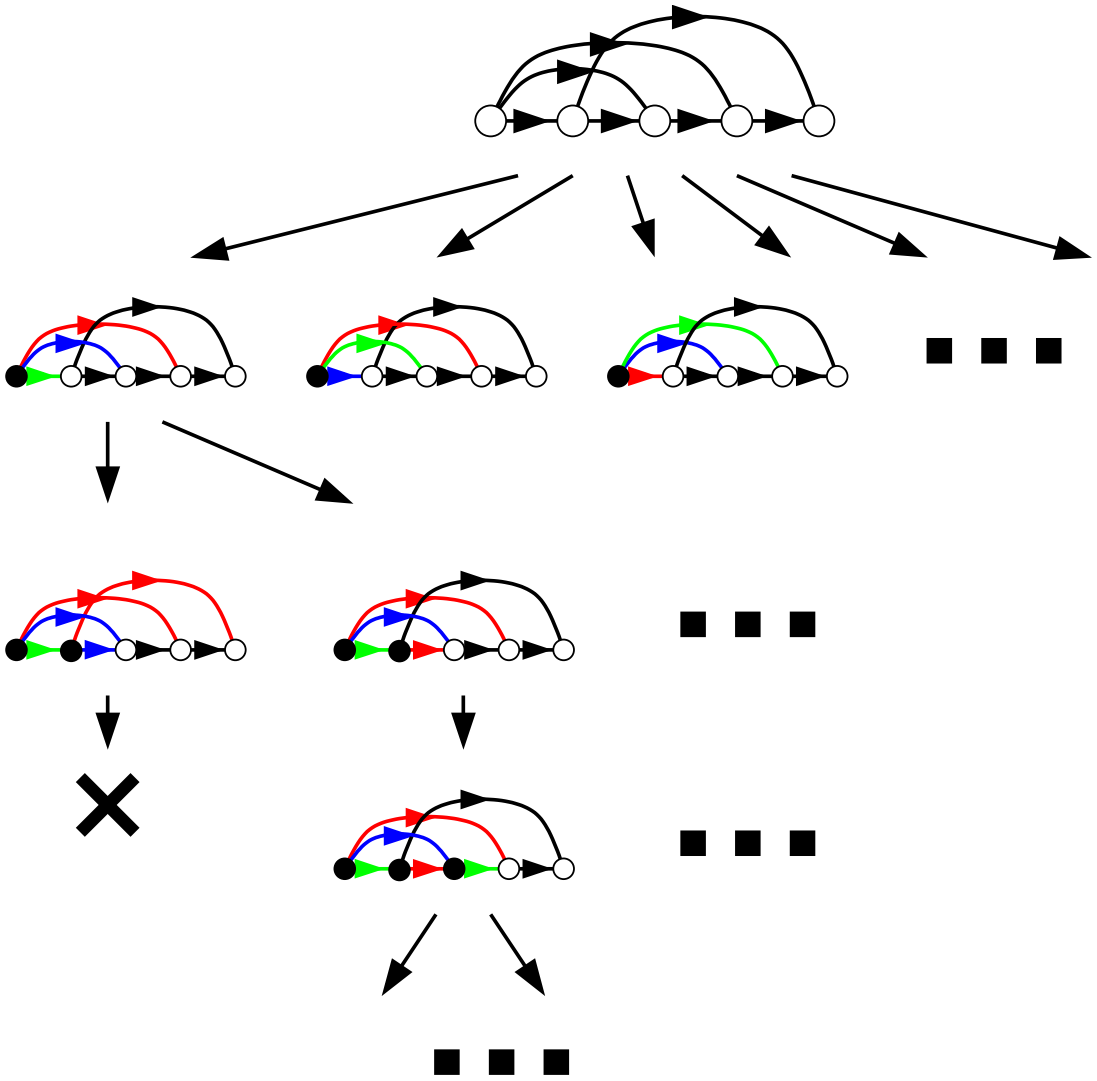
**Corollary (Solving Enum-3-edge coloring:)**

*There exists a branching algorithm with running time* $O^\star(2^{\frac{n}{2}}) = O^\star(1.4143^n)$ *and polynomial space to solve* Enum-3-edge coloring.

**Proof:**

- *If* $\Delta \geq 4$ answer 0.
- If the graph is not connected enough, divide the instance.
- Otherwise, run a branching algorithm.

# Enum-3-edge coloring:



# Extensions


1. Introduction
   - Colorings
   - Algorithmic problems
   - Our results
2. Enumerating the 3-edge colorings of a cubic graph
   - The 3-edge colorings of a 3-regular graph
   - Turning the proof into algorithm
3. Extensions: $k$-edge colorings and the total colorings
   - $k$-edge colorings
   - A more precise bounds for the 3-edge coloring
   - Total coloring
4. Conclusion

# Number of $k$-edge colorings of $k$-regular graphs

We denote by $c_k$ the number of $k$-edge colorings of $G$.

**Theorem**

*Let $G$ be a $k$-regular graph. Then $c_k(G) \leq k \cdot (k-1!)^{\frac{n}{2}}$.*

# Number of $k$-edge colorings of $k$-regular graphs

We denote by $c_k$ the number of $k$-edge colorings of $G$.

**Theorem**

*Let $G$ be a $k$-regular graph. Then $c_k(G) \leq k \cdot (k-1!)^{\frac{n}{2}}$.*

**Proof:** $G$ is 2-connected,

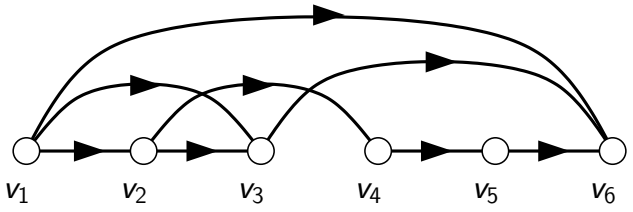
# Number of $k$-edge colorings of $k$-regular graphs

We denote by $c_k$ the number of $k$-edge colorings of $G$.

**Theorem**

*Let $G$ be a $k$-regular graph. Then $c_k(G) \leq k \cdot (k-1!)^{\frac{n}{2}}$.*

**Proof:** $G$ is 2-connected,consider $v_1, \ldots, v_n$ an *st-ordering* of $G$:



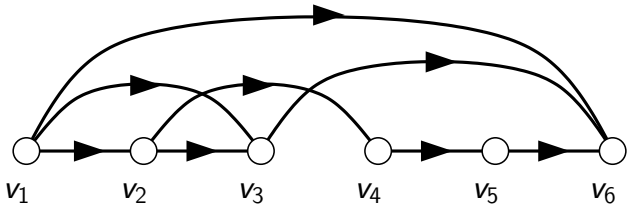
# Number of $k$-edge colorings of $k$-regular graphs

We denote by $c_k$ the number of $k$-edge colorings of $G$.

**Theorem**

*Let $G$ be a $k$-regular graph. Then $c_k(G) \leq k \cdot (k-1!)^{\frac{n}{2}}$.*

**Proof:** $G$ is 2-connected,consider $v_1, \ldots, v_n$ an *st-ordering* of $G$:

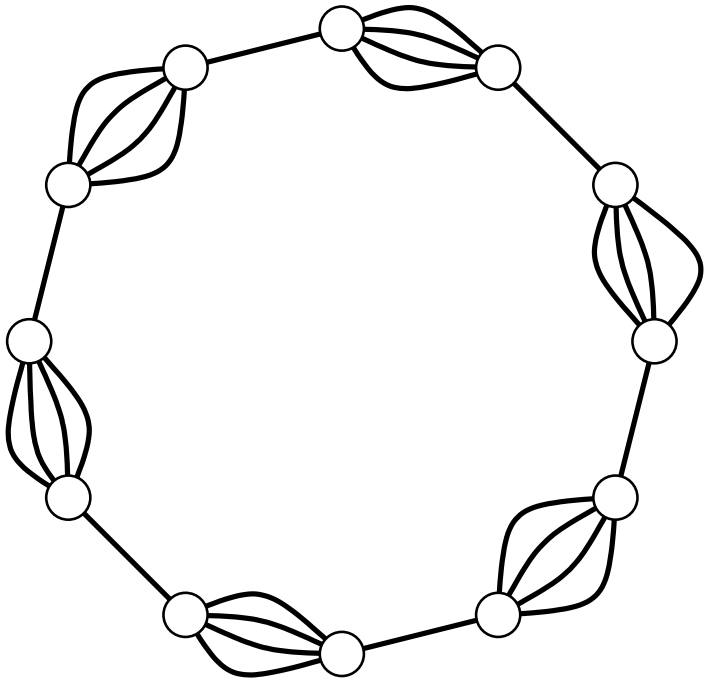



$A_i = \{v \ : \ d^+(v) = i\}$ and $a_i = |A_i|$ for $i = 1, \ldots, k-1$

Then, $c_k(G) \leq \prod_{i=1}^{n-1} d^+(v_i)! \leq k! \prod_{i=1}^{k-1} (i!)^{a_i}$
under $\sum_{i=1}^{k-1} a_i = n-2$ and $\sum_{i=1}^{k-1} i.a_i = k(n-2)/2$ $\square$

# Number of $k$-edge colorings of $k$-regular graphs

Also sharp for multi-graphs:



# Enum-$k$-edge coloring:

**Corollary (Solving Enum-3-edge coloring:)**

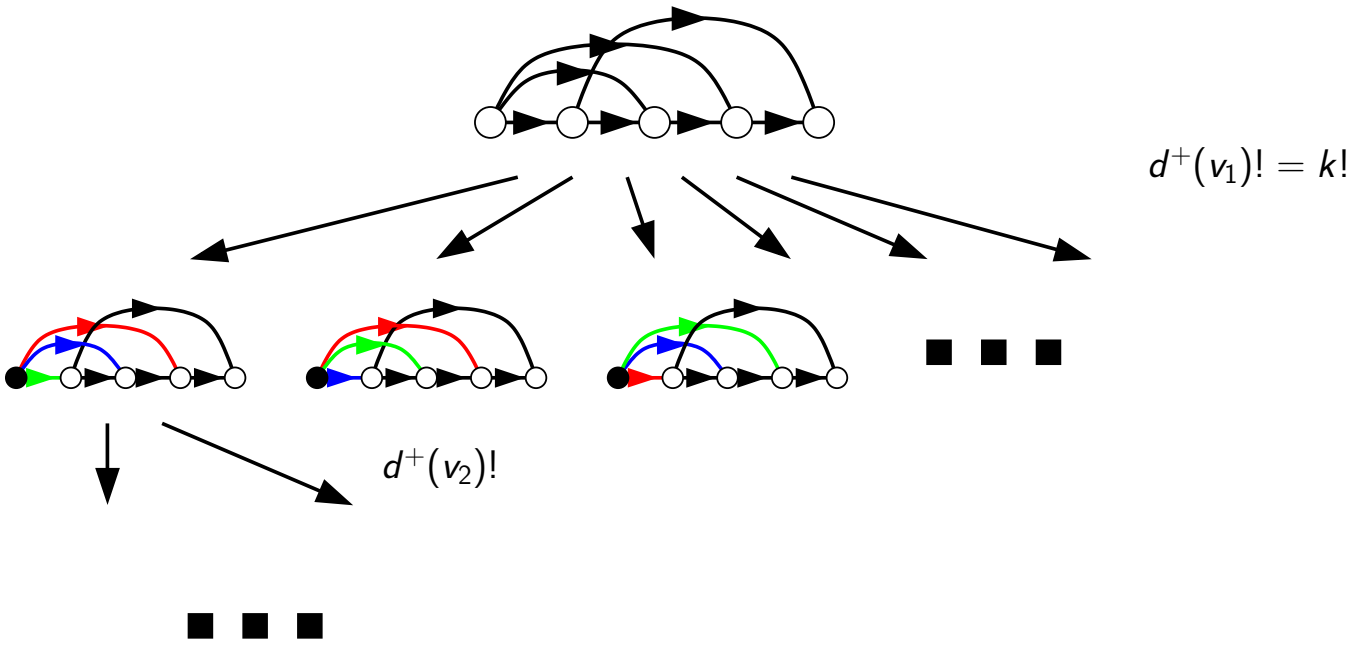
*There exists a branching algorithm with running time $O^\star((k-1!)^{\frac{n}{2}})$ and polynomial space to solve* Enum-$k$-edge coloring.

# Enum-$k$-edge coloring:

**Corollary (Solving Enum-3-edge coloring:)**

*There exists a branching algorithm with running time $O^\star((k-1!)^{\frac{n}{2}})$ and polynomial space to solve Enum-$k$-edge coloring.*

**Proof:**



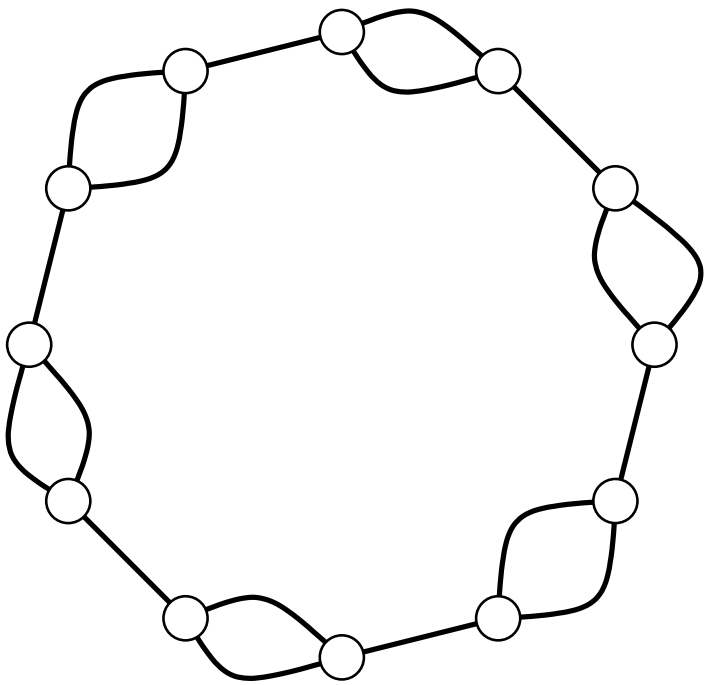

At most $\prod_{i=1}^{n-1} d^+(v_i)! = O^\star((k-1!)^{\frac{n}{2}})$ leaves

# Number of 3-edge colorings for simple graphs

**Corollary**

*Let $G$ be a cubic graph, then $c_3(G) \leq 3 \cdot 2^{\frac{n}{2}}$.*

The sharp example for the number of 3-edge coloring of cubic graphs:

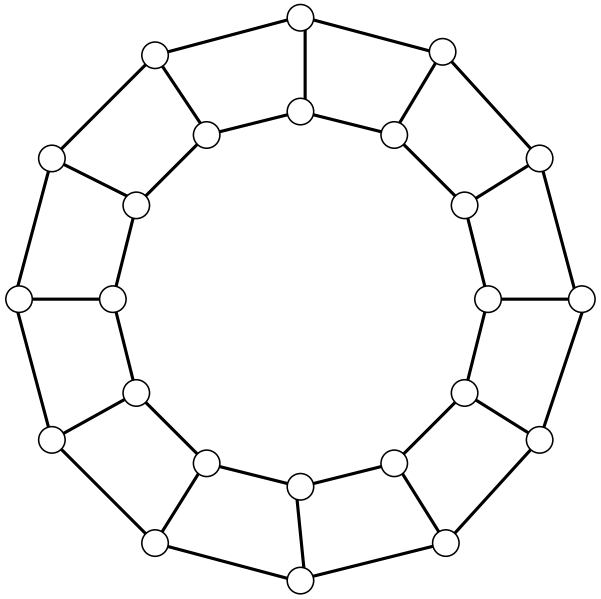


# Number of 3-edge colorings for simple graphs

Can we improve a lot?

# Number of 3-edge colorings for simple graphs

Can we improve a lot? Not really: let $H_n$ be:

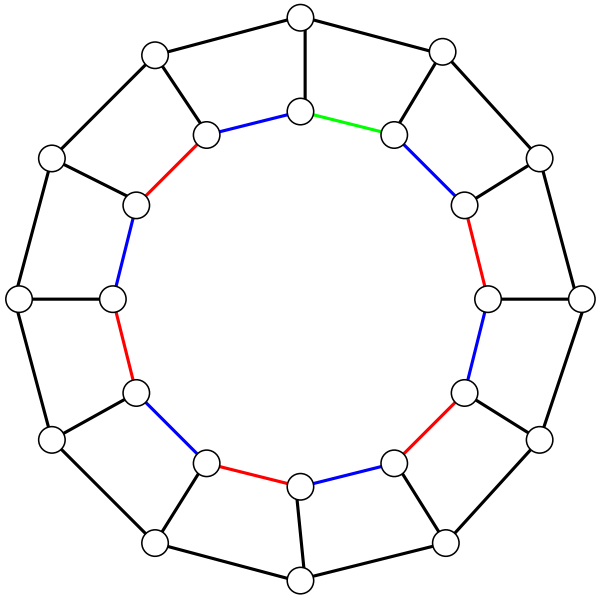


**Lemma ($c_3$ of the ladder graph)**

$$c_3(H_n) = \begin{cases} 2^{n/2} + 8, & \text{if } n/2 \text{ is even,} \\ 2^{n/2} - 2, & \text{if } n/2 \text{ is odd.} \end{cases}$$

# Number of 3-edge colorings for simple graphs

Can we improve a lot? Not really: let $H_n$ be:

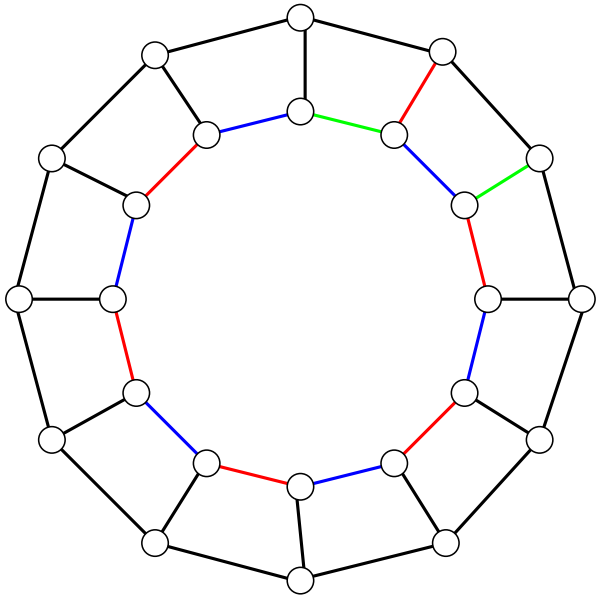


**Lemma ($c_3$ of the ladder graph)**

$$c_3(H_n) = \begin{cases} 2^{n/2} + 8, & \text{if } n/2 \text{ is even,} \\ 2^{n/2} - 2, & \text{if } n/2 \text{ is odd.} \end{cases}$$

# Number of 3-edge colorings for simple graphs

Can we improve a lot? Not really: let $H_n$ be:

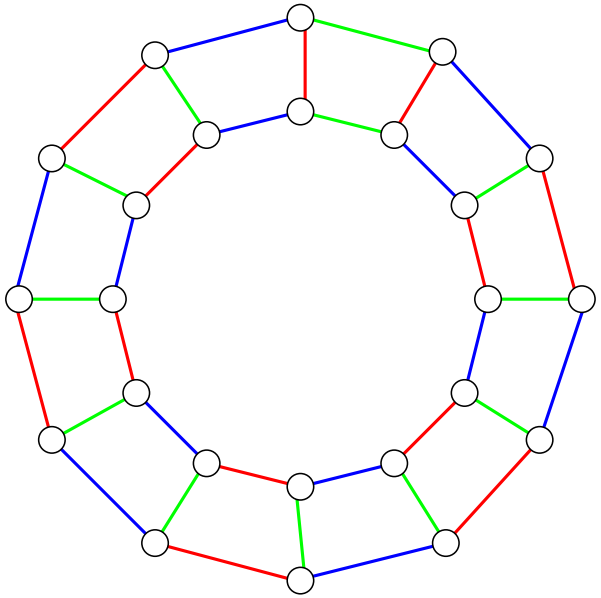


**Lemma ($c_3$ of the ladder graph)**

$$c_3(H_n) = \begin{cases} 2^{n/2} + 8, & \text{if } n/2 \text{ is even,} \\ 2^{n/2} - 2, & \text{if } n/2 \text{ is odd.} \end{cases}$$

# Number of 3-edge colorings for simple graphs

Can we improve a lot? Not really: let $H_n$ be:

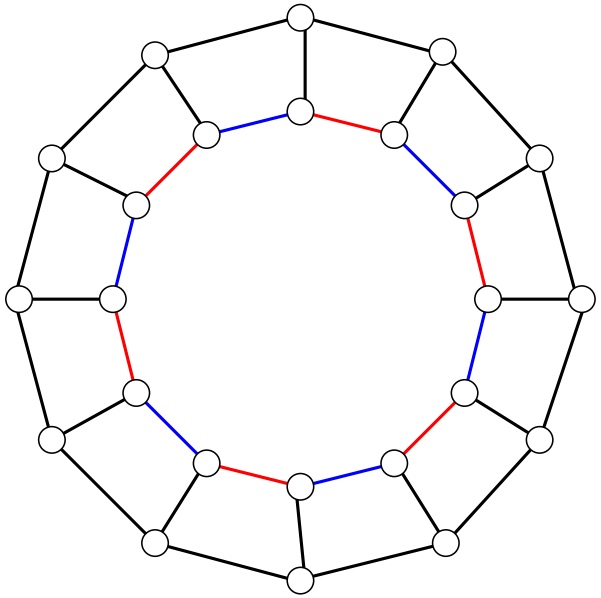


**Lemma ($c_3$ of the ladder graph)**

$$c_3(H_n) = \begin{cases} 2^{n/2} + 8, & \text{if } n/2 \text{ is even,} \\ 2^{n/2} - 2, & \text{if } n/2 \text{ is odd.} \end{cases}$$

# Number of 3-edge colorings for simple graphs

Can we improve a lot? Not really: let $H_n$ be:

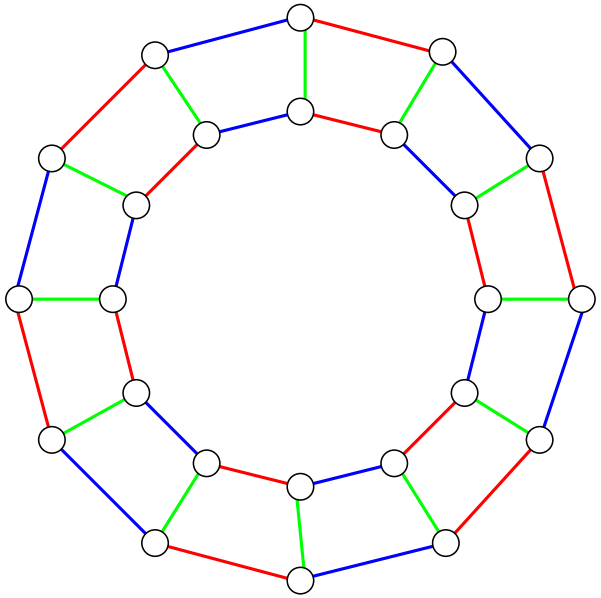


**Lemma ($c_3$ of the ladder graph)**

$$c_3(H_n) = \begin{cases} 2^{n/2} + 8, & \text{if } n/2 \text{ is even,} \\ 2^{n/2} - 2, & \text{if } n/2 \text{ is odd.} \end{cases}$$

# Number of 3-edge colorings for simple graphs

Can we improve a lot? Not really: let $H_n$ be:

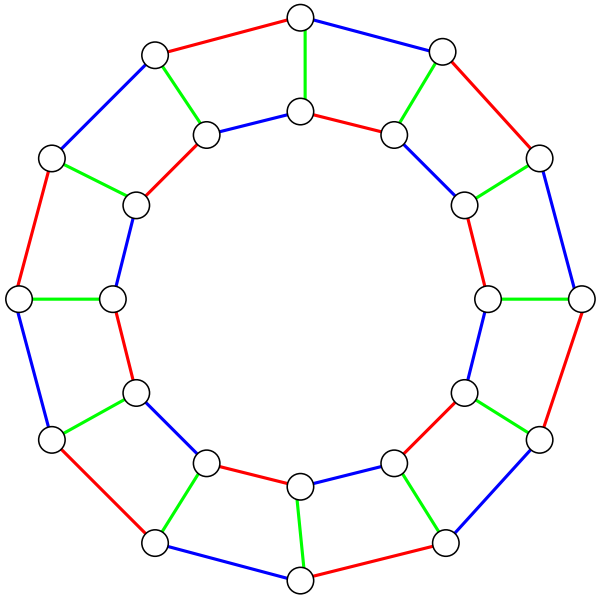


**Lemma ($c_3$ of the ladder graph)**

$$c_3(H_n) = \begin{cases} 2^{n/2} + 8, & \text{if } n/2 \text{ is even,} \\ 2^{n/2} - 2, & \text{if } n/2 \text{ is odd.} \end{cases}$$

# Number of 3-edge colorings for simple graphs

Can we improve a lot? Not really: let $H_n$ be:

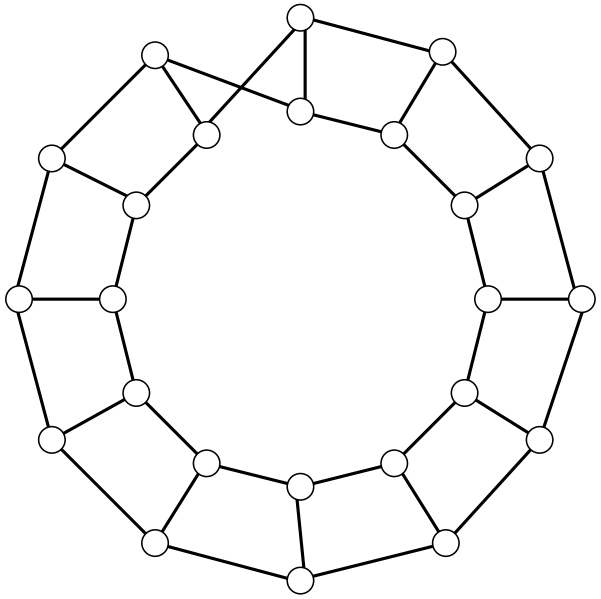


**Lemma ($c_3$ of the ladder graph)**

$$c_3(H_n) = \begin{cases} 2^{n/2} + 8, & \text{if } n/2 \text{ is even,} \\ 2^{n/2} - 2, & \text{if } n/2 \text{ is odd.} \end{cases}$$

# Number of 3-edge colorings for simple graphs

Can we improve a lot? Not really: let $M_n$ be:



**Lemma ($c_3$ of the Mœbius ladder graph)**

$$c_3(M_n) = \begin{cases} 2^{n/2} + 2, & \textit{if } n/2 \textit{ is even,} \\ 2^{n/2} + 4, & \textit{if } n/2 \textit{ is odd.} \end{cases}$$

# Number of 3-edge colorings for simple graphs

So, for simple cubic graphs, we have:

$$2^{\frac{n}{2}} \ < \ \max\{c_3(G) : G \text{ cubic graph on } n \text{ vertices } \} \ \leq \ 3.2^{\frac{n}{2}}$$

# Number of 3-edge colorings for simple graphs

So, for simple cubic graphs, we have:

$$2^{\frac{n}{2}} < \max\{c_3(G) : G \text{ cubic graph on } n \text{ vertices }\} \leq \cancel{3.2^{\frac{n}{2}}} \ \frac{9}{4}.2^{\frac{n}{2}}$$

**Theorem**

*Let $G$ be a connected simple cubic graph. Then $c_3(G) \leq \frac{9}{4} \cdot 2^{n-\frac{n_3}{2}}$.*

# Number of 3-edge colorings for simple graphs

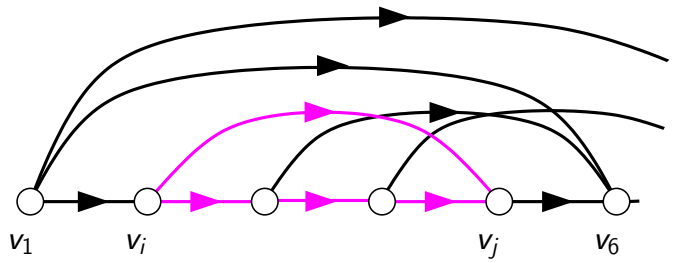
So, for simple cubic graphs, we have:

$$2^{\frac{n}{2}} < \max\{c_3(G) : G \text{ cubic graph on } n \text{ vertices }\} \leq \cancel{3.2^{\frac{n}{2}}} \ \frac{9}{4}.2^{\frac{n}{2}}$$

**Theorem**

*Let $G$ be a connected simple cubic graph. Then $c_3(G) \leq \frac{9}{4} \cdot 2^{n-\frac{n_3}{2}}$.*

**Proof:**




We have $\frac{1}{3} \cdot c_3(C_{j-i+1})/2^{j-i} \cdot 3.2^{\frac{n}{2}}$ colorings.

# 4-total colorings

Let $c_4^T(G)$ be the number of 4-total colorings of $G$.

# 4-total colorings

Let $c_4^T(G)$ be the number of 4-total colorings of $G$.

**Theorem**

*Let $G$ be a (2-)connected cubic graph. Then $c_4^T(G) \leq 3 \cdot 2^{\frac{3n}{2}}$.*

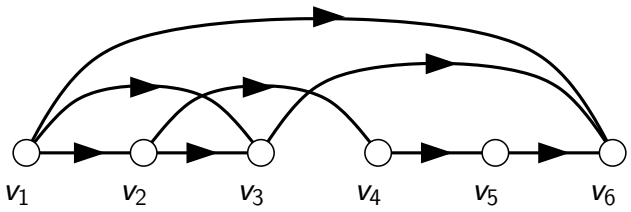
# 4-total colorings

Let $c_4^T(G)$ be the number of 4-total colorings of $G$.

**Theorem**

*Let $G$ be a (2-)connected cubic graph. Then $c_4^T(G) \leq 3 \cdot 2^{\frac{3n}{2}}$.*

**Proof:**




Orient from left to right.
$A^+ = \{v \ : \ d^+(v) = 2 \text{ and } d^-(v) = 1\}$
$A^- = \{v \ : \ d^+(v) = 1 \text{ and } d^-(v) = 2\}$
We have $|A^+| = |A^-| = (n-2)/2$

# 4-total colorings

**Theorem**

*Let $G$ be a (2-)connected cubic graph. Then $c_4^T(G) \leq 3 \cdot 2^{\frac{3n}{2}}$.*

**Proof:** We have $|A^+| = |A^-| = (n-2)/2$
Denote by $c_i$ the number of partial 4-total colorings of vertices and arcs with tail in $\{v_1, \ldots, v_i\}$.

# 4-total colorings

**Theorem**

*Let $G$ be a (2-)connected cubic graph. Then $c_4^T(G) \leq 3 \cdot 2^{\frac{3n}{2}}$.*

**Proof:** We have $|A^+| = |A^-| = (n-2)/2$
Denote by $c_i$ the number of partial 4-total colorings of vertices and arcs with tail in $\{v_1, \ldots, v_i\}$.

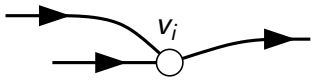
- $c_1 = 24$

# 4-total colorings

**Theorem**

*Let $G$ be a (2-)connected cubic graph. Then $c_4^T(G) \leq 3 \cdot 2^{\frac{3n}{2}}$.*

**Proof:** We have $|A^+| = |A^-| = (n-2)/2$
Denote by $c_i$ the number of partial 4-total colorings of vertices and arcs with tail in $\{v_1, \ldots, v_i\}$.

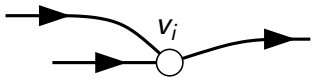
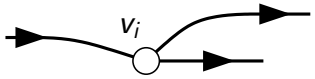
- $c_1 = 24$
- if $v_i \in A^-$  $c_i \leq 2c_{i-1}$

# 4-total colorings

**Theorem**

*Let $G$ be a (2-)connected cubic graph. Then $c_4^T(G) \leq 3 \cdot 2^{\frac{3n}{2}}$.*

**Proof:** We have $|A^+| = |A^-| = (n-2)/2$
Denote by $c_i$ the number of partial 4-total colorings of vertices and arcs with tail in $\{v_1, \ldots, v_i\}$.

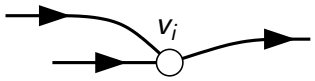
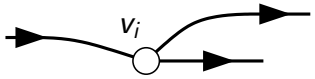
- $c_1 = 24$
- if $v_i \in A^-$  $c_i \leq 2c_{i-1}$
- if $v_i \in A^+$  $c_i \leq 4c_{i-1}$

# 4-total colorings

**Theorem**

*Let $G$ be a (2-)connected cubic graph. Then $c_4^T(G) \leq 3 \cdot 2^{\frac{3n}{2}}$.*

**Proof:** We have $|A^+| = |A^-| = (n-2)/2$
Denote by $c_i$ the number of partial 4-total colorings of vertices and arcs with tail in $\{v_1, \ldots, v_i\}$.

- $c_1 = 24$
- if $v_i \in A^-$  $c_i \leq 2c_{i-1}$
- if $v_i \in A^+$  $c_i \leq 4c_{i-1}$

Hence:
$c_4^T(G) \leq 24 \cdot 2^{\frac{(n-2)}{2}} \cdot 4^{\frac{(n-2)}{2}} = 3 \cdot 2^{\frac{3n}{2}}$ □

# Enum-4-total coloring:

**Corollary (Solving Enum-3-edge coloring:)**

*There exists a branching algorithm with running time* $O^\star(2^{\frac{3n}{2}}) = O^\star(2.8285^n)$ *and polynomial space to solve* Enum-4-total coloring.

# Enum-4-total coloring:

**Corollary (Solving Enum-3-edge coloring:)**

*There exists a branching algorithm with running time* $O^\star(2^{\frac{3n}{2}}) = O^\star(2.8285^n)$ *and polynomial space to solve* Enum-4-total coloring.

# Enum-4-total coloring:

A lower bound on $c_4^T$, the nb of 4-total colorings:

# Enum-4-total coloring:

A lower bound on $c_4^T$, the nb of 4-total colorings:

**Lemma**

*Let $T$ be a binary tree (degree = 1 or 3) on $n$ vertices, $c_4^T(T) = 3 \cdot 2^{\frac{3n}{2}}$.*

# Enum-4-total coloring:

A lower bound on $c_4^T$, the nb of 4-total colorings:

**Lemma**

*Let $T$ be a binary tree (degree = 1 or 3) on $n$ vertices, $c_4^T(T) = 3 \cdot 2^{\frac{3n}{2}}$.*

**Proof:** by induction:

# Enum-4-total coloring:

A lower bound on $c_4^T$, the nb of 4-total colorings:

**Lemma**

*Let $T$ be a binary tree (degree = 1 or 3) on $n$ vertices, $c_4^T(T) = 3 \cdot 2^{\frac{3n}{2}}$.*

**Proof:** by induction:

- $c_4^T(K_{1,3}) = 4.3!.2^3 = 3.2^6 = 3.2^{\frac{3.4}{2}}$

# Enum-4-total coloring:

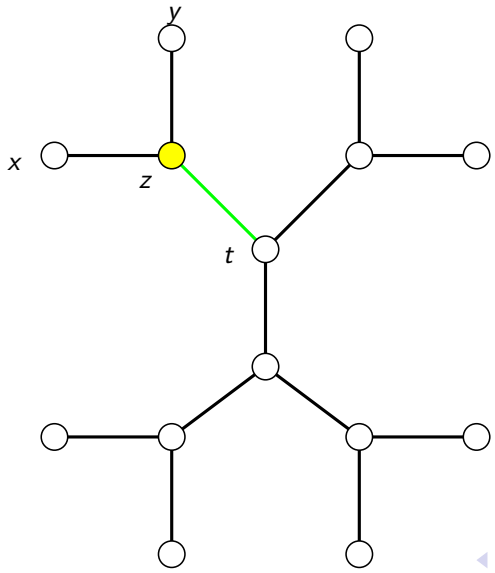
A lower bound on $c_4^T$, the nb of 4-total colorings:

**Lemma**

*Let $T$ be a binary tree (degree = 1 or 3) on $n$ vertices, $c_4^T(T) = 3 \cdot 2^{\frac{3n}{2}}$.*

**Proof:** by induction:

- $c_4^T(K_{1,3}) = 4.3!.2^3 = 3.2^6 = 3.2^{\frac{3.4}{2}}$
- $c_4^T(T) = 8.c_4^T(T \setminus \{x, y\})$



# Enum-4-total coloring:

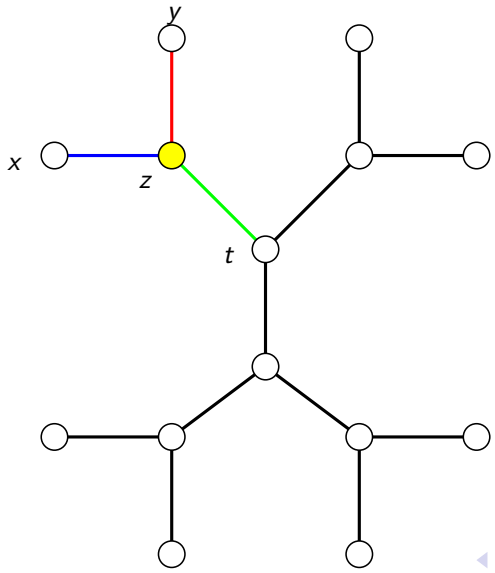
A lower bound on $c_4^T$, the nb of 4-total colorings:

**Lemma**

*Let $T$ be a binary tree (degree = 1 or 3) on $n$ vertices, $c_4^T(T) = 3 \cdot 2^{\frac{3n}{2}}$.*

**Proof:** by induction:

- $c_4^T(K_{1,3}) = 4.3!.2^3 = 3.2^6 = 3.2^{\frac{3.4}{2}}$
- $c_4^T(T) = 8.c_4^T(T \setminus \{x, y\})$

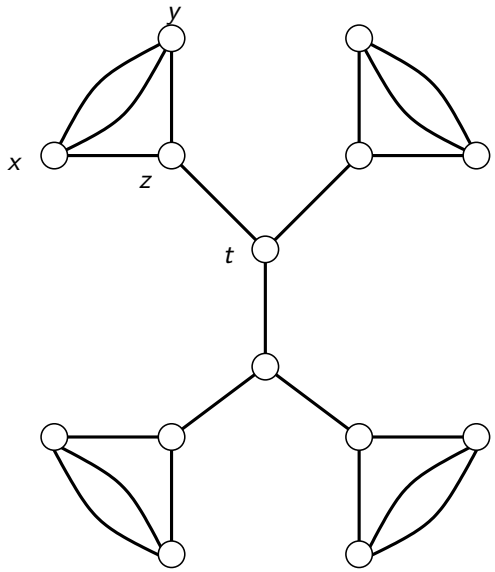


# Enum-4-total coloring:

**Lemma**

*Let $T_n$ be a binary tree on $n$ vertices + parallel edges between twin leaves,* $c_4^T(T_n) = \frac{3}{\sqrt{2}} \cdot 2^{\frac{5n}{4}}$.

**Proof:**



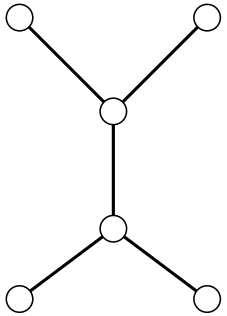

$n = p - 2 + p + 2p$

# Enum-4-total coloring:

**Lemma**

*Let $T_n$ be a binary tree on $n$ vertices + parallel edges between twin leaves,* $c_4^T(T_n) = \frac{3}{\sqrt{2}} \cdot 2^{\frac{5n}{4}}$.

**Proof:**



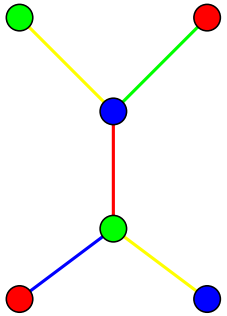
$c_4^T(T') = 3 \cdot 2^{\frac{3(2p-2)}{2}}$

# Enum-4-total coloring:

**Lemma**

*Let $T_n$ be a binary tree on $n$ vertices + parallel edges between twin leaves,* $c_4^T(T_n) = \frac{3}{\sqrt{2}} \cdot 2^{\frac{5n}{4}}$.

**Proof:**



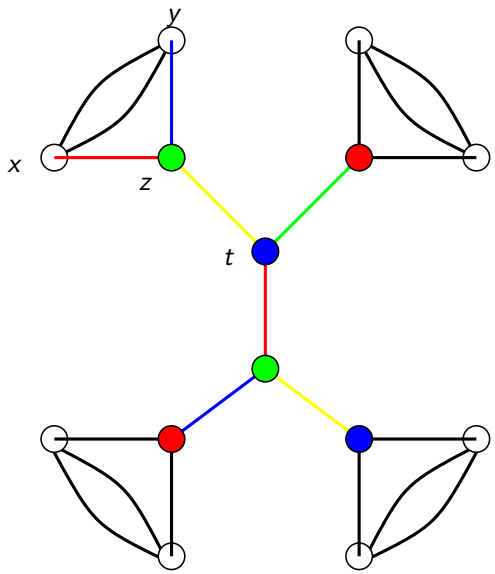
$c_4^T(T') = 3 \cdot 2^{\frac{3(2p-2)}{2}}$

# Enum-4-total coloring:

**Lemma**

*Let $T_n$ be a binary tree on $n$ vertices + parallel edges between twin leaves,* $c_4^T(T_n) = \frac{3}{\sqrt{2}} \cdot 2^{\frac{5n}{4}}$.

**Proof:**



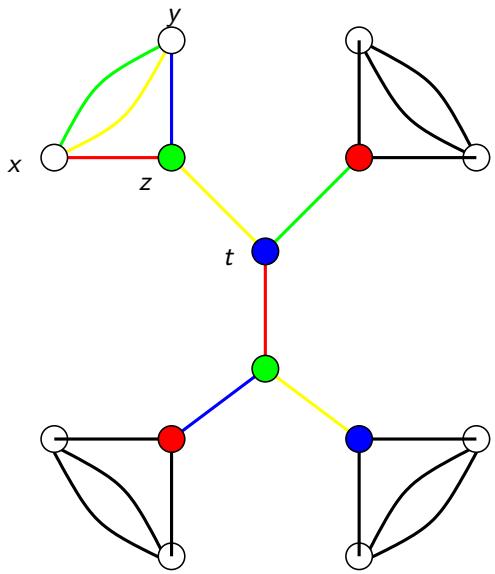

$$c_4^T(T') = 3 \cdot 2^{\frac{3(2p-2)}{2}} \cdot 4^p$$

# Enum-4-total coloring:

**Lemma**

*Let $T_n$ be a binary tree on $n$ vertices + parallel edges between twin leaves,* $c_4^T(T_n) = \frac{3}{\sqrt{2}} \cdot 2^{\frac{5n}{4}}$.

**Proof:**




$$c_4^T(T') = 3 \cdot 2^{\frac{3(2p-2)}{2}} \cdot 4^p = 3 \cdot 2^{5p-3} = 3 \cdot 2^{\frac{5n}{4}-\frac{1}{2}}$$

□

# Conclusion


1. Introduction
   - Colorings
   - Algorithmic problems
   - Our results
2. Enumerating the 3-edge colorings of a cubic graph
   - The 3-edge colorings of a 3-regular graph
   - Turning the proof into algorithm
3. Extensions: $k$-edge colorings and the total colorings
   - $k$-edge colorings
   - A more precise bounds for the 3-edge coloring
   - Total coloring
4. Conclusion

# Some (hard?) open questions

- Minimum vertex coloring in $O^\star(2^n)$ (A. Bjöklund, T. Husfeld and M. Koivisto, 2006)

# Some (hard?) open questions

- Minimum vertex coloring in $O^\star(2^n)$ (A. Bjöklund, T. Husfeld and M. Koivisto, 2006)
  Is there a $O^\star((2-\epsilon)^n)$ algorithm for some $\epsilon > 0$

# Some (hard?) open questions

- Minimum vertex coloring in $O^\star(2^n)$ (A. Bjöklund, T. Husfeld and M. Koivisto, 2006)
  Is there a $O^\star((2-\epsilon)^n)$ algorithm for some $\epsilon > 0$

- Minimum edge coloring in $O^\star(2^m)$

# Some (hard?) open questions

- Minimum vertex coloring in $O^\star(2^n)$ (A. Bjöklund, T. Husfeld and M. Koivisto, 2006)
  Is there a $O^\star((2-\epsilon)^n)$ algorithm for some $\epsilon > 0$

- Minimum edge coloring in $O^\star(2^m)$
  Is there a $O^\star(2^n)$ algorithm

# Some (hard?) open questions

- Minimum vertex coloring in $O^\star(2^n)$ (A. Bjöklund, T. Husfeld and M. Koivisto, 2006)
  Is there a $O^\star((2-\epsilon)^n)$ algorithm for some $\epsilon > 0$

- Minimum edge coloring in $O^\star(2^m)$
  Is there a $O^\star(2^n)$ algorithm

- 3-vertex coloring in $O^\star(1.3289^n)$ (R. Beigel and D. Eppstein, 2005)

# Some (hard?) open questions

- Minimum vertex coloring in $O^\star(2^n)$ (A. Bjöklund, T. Husfeld and M. Koivisto, 2006)
  Is there a $O^\star((2-\epsilon)^n)$ algorithm for some $\epsilon > 0$

- Minimum edge coloring in $O^\star(2^m)$
  Is there a $O^\star(2^n)$ algorithm

- 3-vertex coloring in $O^\star(1.3289^n)$ (R. Beigel and D. Eppstein, 2005)
  Is it possible to do better?

# Some (hard?) open questions

- MINIMUM VERTEX COLORING in $O^\star(2^n)$ (A. Bjöklund, T. Husfeld and M. Koivisto, 2006)
  Is there a $O^\star((2-\epsilon)^n)$ algorithm for some $\epsilon > 0$

- MINIMUM EDGE COLORING in $O^\star(2^m)$
  Is there a $O^\star(2^n)$ algorithm

- 3-VERTEX COLORING in $O^\star(1.3289^n)$ (R. Beigel and D. Eppstein, 2005)
  Is it possible to do better?
  Under ETH, there is no $O^\star(2^{o(n)})$ algorithm for 3-VERTEX COLORING (D. Lokshtanov, D. Marx, S. Saurabh, 2012), several scenarios...

# Some (less hard?) open questions

- Enum-3-edge coloring in $O^\star(1.4142^n)$ and a sharp example (multi-graph).

# Some (less hard?) open questions

- Enum-3-edge coloring in $O^\star(1.4142^n)$ and a sharp example (multi-graph).
  Is it possible to fix
  $c_3(n) = \max\{c_3(G) \ : \ G \text{ simple graph on } n \text{ vertices}\}$
  We know that $2^{\frac{n}{2}} < c_3(n) \leq \frac{9}{4} \cdot 2^{\frac{n}{2}}$

# Some (less hard?) open questions

- Enum-3-edge coloring in $O^\star(1.4142^n)$ and a sharp example (multi-graph).
  Is it possible to fix
  $c_3(n) = \max\{c_3(G) \ : \ G \text{ simple graph on } n \text{ vertices}\}$
  We know that $2^{\frac{n}{2}} < c_3(n) \leq \frac{9}{4} \cdot 2^{\frac{n}{2}}$

- Enum-4-total coloring in $O^\star(2.8285^n)$. Example in $\Omega(2.3784^n)$.

# Some (less hard?) open questions

- Enum-3-edge coloring in $O^\star(1.4142^n)$ and a sharp example (multi-graph).
  Is it possible to fix
  $c_3(n) = \max\{c_3(G) \ : \ G \text{ simple graph on } n \text{ vertices}\}$
  We know that $2^{\frac{n}{2}} < c_3(n) \leq \frac{9}{4} \cdot 2^{\frac{n}{2}}$

- Enum-4-total coloring in $O^\star(2.8285^n)$. Example in $\Omega(2.3784^n)$.
  Is it possible to find the algorithm with best running time for Enum-4-total coloring.

Thank you for your attention !