One-Two Trees let's twist!

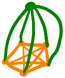
Julien COURTIEL, Matthieu DIEN, Paul DORBEC

PART I

WHY CHORDAL GRAPHS?

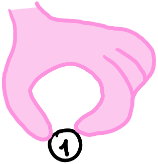
WHAT IS A CHORDAL GRAPH?

DEFINITION

chordal graph =
- start from
- pick  CLIQUE → attach new vertex 
- repeat

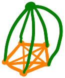
clique = subgraph where every pair of vertices is connected.

Ex:



WHAT IS A CHORDAL GRAPH?

DEFINITION

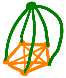
chordal graph =
- start from 
- pick  CLIQUE → attach new vertex 
- repeat

clique = subgraph where every pair of vertices is connected.

Ex:

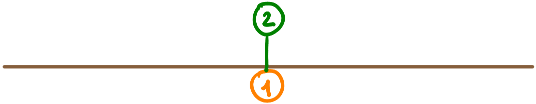
WHAT IS A CHORDAL GRAPH?

DEFINITION

chordal graph =
- start from •
- pick  CLIQUE → attach new vertex 
- repeat

clique = subgraph where every pair of vertices is connected.

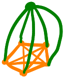
Ex:



WHAT IS A CHORDAL GRAPH?

DEFINITION

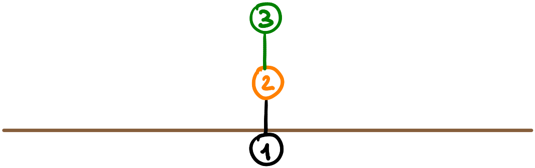
chordal graph = • start from ●

• pick  CLIQUE → attach new vertex 

• repeat

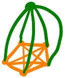
clique = subgraph where every pair of vertices is connected.

Ex:



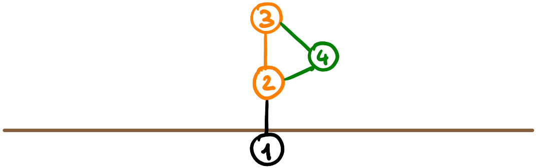
WHAT IS A CHORDAL GRAPH?

DEFINITION

chordal graph =
- start from ●
- pick  CLIQUE → attach new vertex 
- repeat

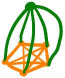
clique = subgraph where every pair of vertices is connected.

Ex:



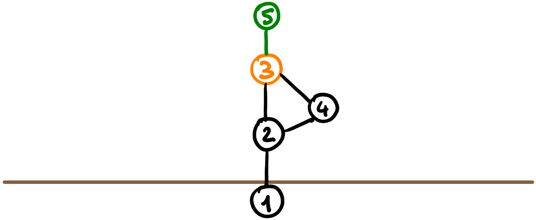
WHAT IS A CHORDAL GRAPH?

DEFINITION

chordal graph =
- start from ●
- pick  CLIQUE → attach new vertex 
- repeat

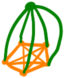
clique = subgraph where every pair of vertices is connected.

Ex:



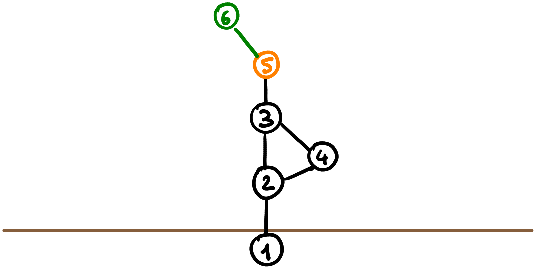
WHAT IS A CHORDAL GRAPH?

DEFINITION

chordal graph =
- start from ●
- pick  CLIQUE → attach new vertex 
- repeat

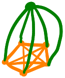
clique = subgraph where every pair of vertices is connected.

Ex:



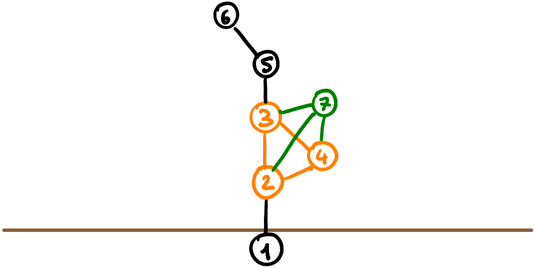
WHAT IS A CHORDAL GRAPH?

DEFINITION

chordal graph =
- start from ●
- pick  CLIQUE → attach new vertex 
- repeat

clique = subgraph where every pair of vertices is connected.

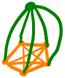
Ex:



WHAT IS A CHORDAL GRAPH?

DEFINITION

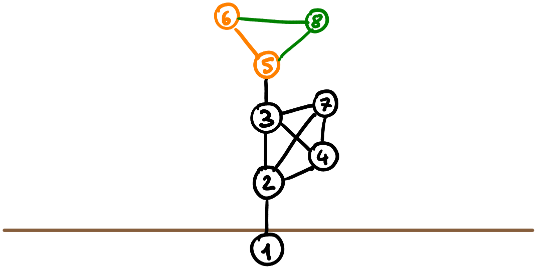
chordal graph = • start from 

• pick  CLIQUE → attach new vertex 

• repeat

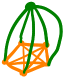
clique = subgraph where every pair of vertices is connected.

Ex:



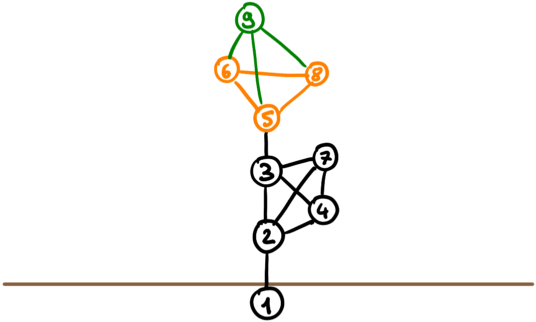
WHAT IS A CHORDAL GRAPH?

DEFINITION

chordal graph =
- start from ●
- pick  CLIQUE → attach new vertex 
- repeat

clique = subgraph where every pair of vertices is connected.

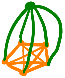
Ex:



WHAT IS A CHORDAL GRAPH?

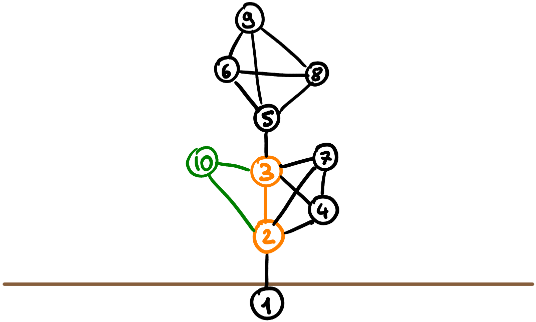
DEFINITION

chordal graph =

- start from 
- pick 
  CLIQUE → attach new vertex 
- repeat

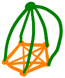
clique = subgraph where every pair of vertices is connected.

Ex:



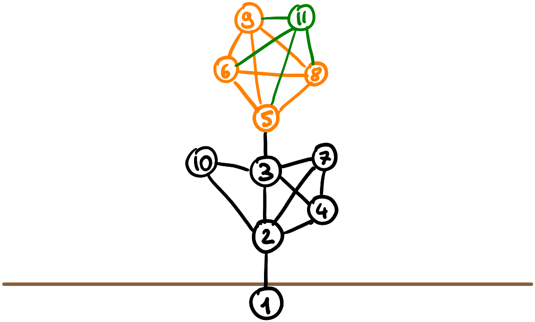
WHAT IS A CHORDAL GRAPH?

DEFINITION

chordal graph =
- start from •
- pick  CLIQUE → attach new vertex 
- repeat

clique = subgraph where every pair of vertices is connected.

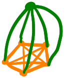
Ex:



WHAT IS A CHORDAL GRAPH?

DEFINITION

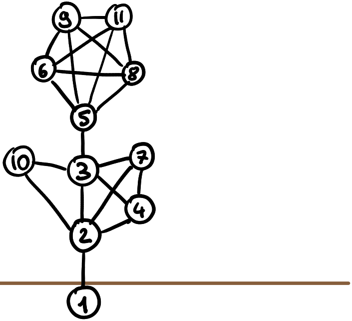
chordal graph = • start from •

• pick  CLIQUE → attach new vertex 

• repeat ↺

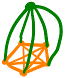
clique = subgraph where every pair of vertices is connected.

Ex:



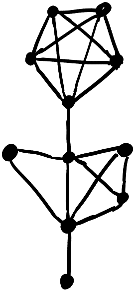
WHAT IS A CHORDAL GRAPH?

DEFINITION

chordal graph =
- start from 
- pick  CLIQUE → attach new vertex 
- repeat

clique = subgraph where every pair of vertices is connected.

Ex:



BACKWARDS DEFINITION

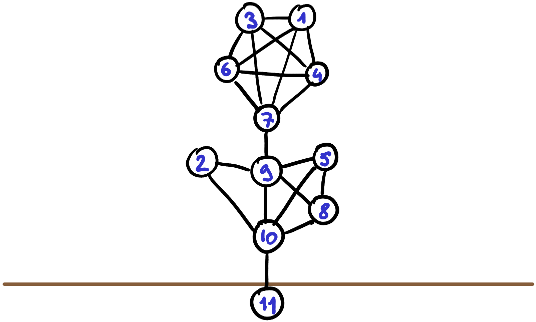
removable vertex = vertex whose neighbors form a clique

perfect elimination order = order that removes all vertices

DEFINITION 2

chordal graph = graph with a perfect elimination order

Ex:



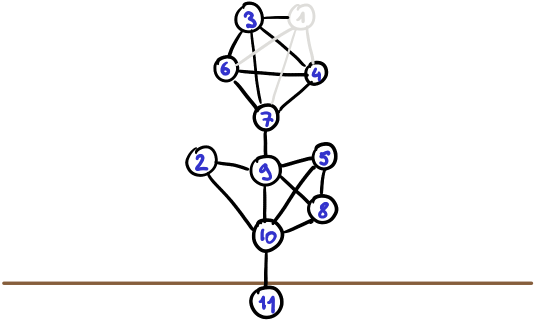
BACKWARDS DEFINITION

removable vertex = vertex whose neighbors form a clique

perfect elimination order = order that removes all vertices

DEFINITION 2
chordal graph = graph with a perfect elimination order

Ex:



BACKWARDS DEFINITION

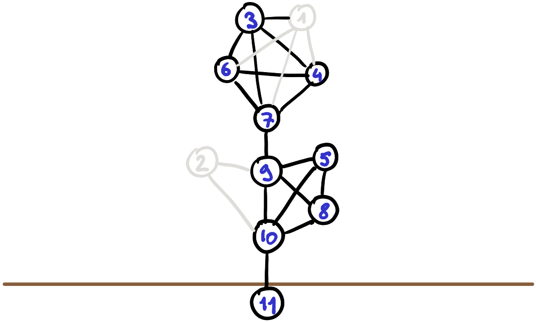
removable vertex = vertex whose neighbors form a clique

perfect elimination order = order that removes all vertices

DEFINITION 2

chordal graph = graph with a perfect elimination order

Ex:



BACKWARDS DEFINITION

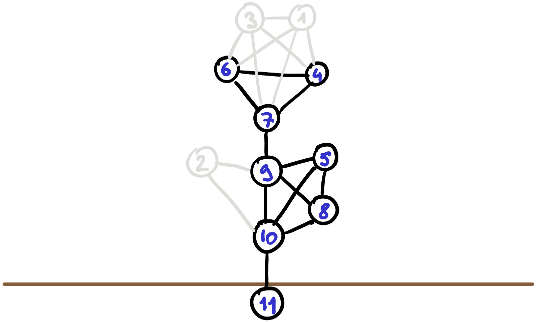
removable vertex = vertex whose neighbors form a clique

perfect elimination order = order that removes all vertices

DEFINITION 2

chordal graph = graph with a perfect elimination order

Ex:



BACKWARDS DEFINITION

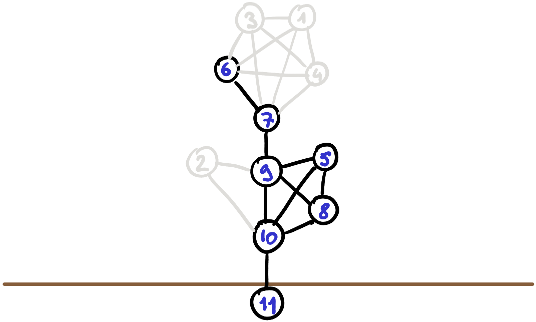
removable vertex = vertex whose neighbors form a clique

perfect elimination order = order that removes all vertices

DEFINITION 2

chordal graph = graph with a perfect elimination order

Ex:



BACKWARDS DEFINITION

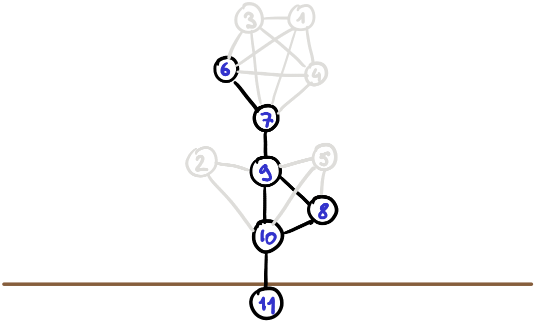
removable vertex = vertex whose neighbors form a clique

perfect elimination order = order that removes all vertices

DEFINITION 2

chordal graph = graph with a perfect elimination order

Ex:



BACKWARDS DEFINITION

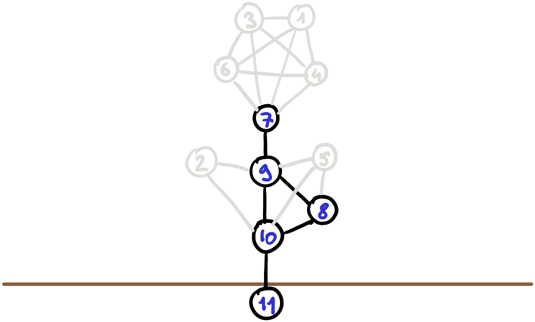
removable vertex = vertex whose neighbors form a clique

perfect elimination order = order that removes all vertices

DEFINITION 2

chordal graph = graph with a perfect elimination order

Ex:



BACKWARDS DEFINITION

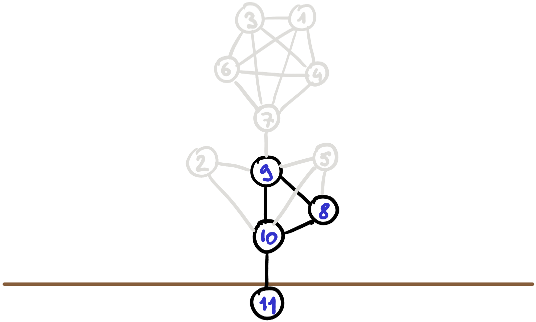
removable vertex = vertex whose neighbors form a clique

perfect elimination order = order that removes all vertices

DEFINITION 2

chordal graph = graph with a perfect elimination order

Ex:



BACKWARDS DEFINITION

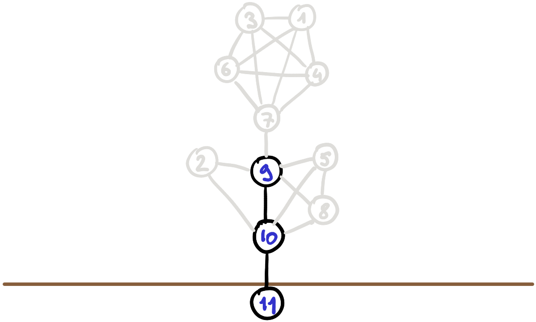
removable vertex = vertex whose neighbors form a clique

perfect elimination order = order that removes all vertices

DEFINITION 2

chordal graph = graph with a perfect elimination order

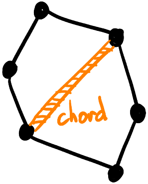
Ex:



FORBIDDEN PATTERN DEFINITION

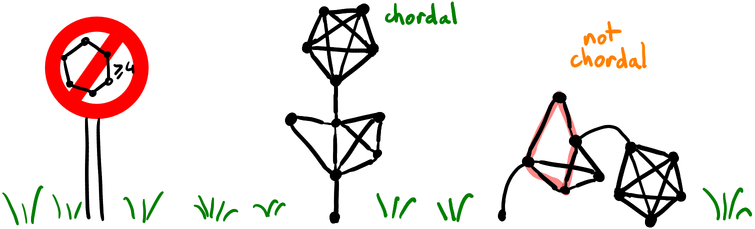
In a cycle,

chord = edge between two non-consecutive vertices



DEFINITION 3

chordal graph = graph without chordless cycles of length ≥ 4.



MOTIVATION BEHIND CHORDAL GRAPHS

$\sim$ 16 000 papers on Google Scholar

(point of comparison: "Turing degree" $\sim$ 4000 papers)

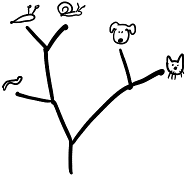
MOTIVATION BEHIND CHORDAL GRAPHS

~ 16 000 papers on Google Scholar
(point of comparison : "Turing degree" ~ 4000 papers)

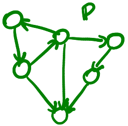
DOMAINS WHERE THEY APPEAR (OUTSIDE THE WORLD OF GRAPHS)

numerical linear algebra

phylogenetics



Bayesian networks



MOTIVATION BEHIND CHORDAL GRAPHS

~ 16 000 papers on Google Scholar

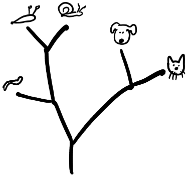
(point of comparison : "Turing degree" ~ 4000 papers)

**DOMAINS WHERE THEY APPEAR
(OUTSIDE THE WORLD OF GRAPHS)**

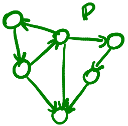
numerical linear algebra

scientific zoom

phylogenetics



Bayesian networks



CHOLESKY DECOMPOSITION

Goal of Cholesky decomposition	

Cholesky decomposition

Goal of Cholesky decomposition	Given a symmetric positive-definite matrix S, find a lower triangular matrix L such that $S = L \, L^T$ $\begin{pmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{pmatrix} = \begin{pmatrix} \times & 0 & 0 \\ \times & \times & 0 \\ \times & \times & \times \end{pmatrix} \begin{pmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \end{pmatrix}$

Cholesky decomposition

Goal of Cholesky decomposition

Given a symmetric positive-definite matrix S, find a lower triangular matrix L such that

$$S = L\,L^T$$

$$\begin{pmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{pmatrix} = \begin{pmatrix} \times & 0 & 0 \\ \times & \times & 0 \\ \times & \times & \times \end{pmatrix}\begin{pmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \end{pmatrix}$$

Remark: Not all decompositions are equivalently good.

Ex:

$$\begin{pmatrix} 5 & 1 & 1 & 1 & 1 \\ 1 & 4 & 0 & 0 & 0 \\ 1 & 0 & 3 & 0 & 0 \\ 1 & 0 & 0 & 2 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2.236 & 0 & 0 & 0 & 0 \\ 0.447 & 1.949 & 0 & 0 & 0 \\ 0.447 & -0.103 & 1.670 & 0 & 0 \\ 0.447 & -0.103 & -0.126 & 1.332 & 0 \\ 0.447 & -0.103 & -0.126 & -0.170 & 0.863 \end{pmatrix}\begin{pmatrix} 2.236 & 0.447 & 0.447 & 0.447 & 0.447 \\ 0 & 1.949 & -0.103 & -0.103 & -0.103 \\ 0 & 0 & 1.670 & -0.126 & -0.126 \\ 0 & 0 & 0 & 1.332 & -0.170 \\ 0 & 0 & 0 & 0 & 0.863 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 3 & 0 & 1 \\ 0 & 0 & 0 & 4 & 1 \\ 1 & 1 & 1 & 1 & 5 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1.414 & 0 & 0 & 0 \\ 0 & 0 & 1.732 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 1 & 0.707 & 0.577 & 0.5 & 1.708 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1.414 & 0 & 0 & 0.707 \\ 0 & 0 & 1.732 & 0 & 0.577 \\ 0 & 0 & 0 & 2 & 0.5 \\ 0 & 0 & 0 & 0 & 1.708 \end{pmatrix}$$

CHOLESKY DECOMPOSITION

Goal of Cholesky decomposition	Given a symmetric positive-definite matrix S, find a lower triangular matrix L such that $$S = L\,L^T$$ $$\begin{pmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{pmatrix} = \begin{pmatrix} \times & 0 & 0 \\ \times & \times & 0 \\ \times & \times & \times \end{pmatrix}\begin{pmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \end{pmatrix}$$

Remark: Not all decompositions are equivalently good.

Ex:

$$\begin{pmatrix} 5 & 1 & 1 & 1 & 1 \\ 1 & 4 & 0 & 0 & 0 \\ 1 & 0 & 3 & 0 & 0 \\ 1 & 0 & 0 & 2 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2.236 & 0 & 0 & 0 & 0 \\ 0.447 & 1.949 & 0 & 0 & 0 \\ 0.447 & -0.103 & 1.670 & 0 & 0 \\ 0.447 & -0.103 & -0.126 & 1.332 & 0 \\ 0.447 & -0.103 & -0.126 & -0.170 & 0.863 \end{pmatrix}\begin{pmatrix} 2.236 & 0.447 & 0.447 & 0.447 & 0.447 \\ 0 & 1.949 & -0.103 & -0.103 & -0.103 \\ 0 & 0 & 1.670 & -0.126 & -0.126 \\ 0 & 0 & 0 & 1.332 & -0.170 \\ 0 & 0 & 0 & 0 & 0.863 \end{pmatrix}$$

same, up to a permutation of coordinates

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 3 & 0 & 1 \\ 0 & 0 & 0 & 4 & 1 \\ 1 & 1 & 1 & 1 & 5 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1.414 & 0 & 0 & 0 \\ 0 & 0 & 1.732 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 1 & 0.707 & 0.577 & 0.5 & 1.708 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1.414 & 0 & 0 & 0.707 \\ 0 & 0 & 1.732 & 0 & 0.577 \\ 0 & 0 & 0 & 2 & 0.5 \\ 0 & 0 & 0 & 0 & 1.708 \end{pmatrix}$$

Cholesky decomposition

Goal of Cholesky decomposition	Given a symmetric positive-definite matrix S, find a lower triangular matrix L such that $S = L L^T$

$$\begin{pmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{pmatrix} = \begin{pmatrix} \times & 0 & 0 \\ \times & \times & 0 \\ \times & \times & \times \end{pmatrix}\begin{pmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \end{pmatrix}$$

Remark: Not all decompositions are equivalently good.

Ex:

$$\begin{pmatrix} 5 & 1 & 1 & 1 & 1 \\ 1 & 4 & 0 & 0 & 0 \\ 1 & 0 & 3 & 0 & 0 \\ 1 & 0 & 0 & 2 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2.236 & 0 & 0 & 0 & 0 \\ 0.447 & 1.949 & 0 & 0 & 0 \\ 0.447 & -0.103 & 1.670 & 0 & 0 \\ 0.447 & -0.103 & -0.126 & 1.332 & 0 \\ 0.447 & -0.103 & -0.126 & -0.170 & 0.863 \end{pmatrix}\begin{pmatrix} 2.236 & 0.447 & 0.447 & 0.447 & 0.447 \\ 0 & 1.949 & -0.103 & -0.103 & -0.103 \\ 0 & 0 & 1.670 & -0.126 & -0.126 \\ 0 & 0 & 0 & 1.332 & -0.170 \\ 0 & 0 & 0 & 0 & 0.863 \end{pmatrix}$$

becomes dense

same, up to a permutation of coordinates

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 3 & 0 & 1 \\ 0 & 0 & 0 & 4 & 1 \\ 1 & 1 & 1 & 1 & 5 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1.414 & 0 & 0 & 0 \\ 0 & 0 & 1.732 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 1 & 0.707 & 0.577 & 0.5 & 1.708 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1.414 & 0 & 0 & 0.707 \\ 0 & 0 & 1.732 & 0 & 0.577 \\ 0 & 0 & 0 & 2 & 0.5 \\ 0 & 0 & 0 & 0 & 1.708 \end{pmatrix}$$

remains sparse

Cholesky decomposition

Goal of Cholesky decomposition	Given a symmetric positive-definite matrix S, find a lower triangular matrix L such that $S = L L^T$

$$\begin{pmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{pmatrix} = \begin{pmatrix} \times & 0 & 0 \\ \times & \times & 0 \\ \times & \times & \times \end{pmatrix} \begin{pmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \end{pmatrix}$$

Remark: Not all decompositions are equivalently good.

Ex:

$$\begin{pmatrix} 5 & 1 & 1 & 1 & 1 \\ 1 & 4 & 0 & 0 & 0 \\ 1 & 0 & 3 & 0 & 0 \\ 1 & 0 & 0 & 2 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2.236 & 0 & 0 & 0 & 0 \\ 0.447 & 1.949 & 0 & 0 & 0 \\ 0.447 & -0.103 & 1.670 & 0 & 0 \\ 0.447 & -0.103 & -0.126 & 1.332 & 0 \\ 0.447 & -0.103 & -0.126 & -0.170 & 0.863 \end{pmatrix} \begin{pmatrix} 2.236 & 0.447 & 0.447 & 0.447 & 0.447 \\ 0 & 1.949 & -0.103 & -0.103 & -0.103 \\ 0 & 0 & 1.670 & -0.126 & -0.126 \\ 0 & 0 & 0 & 1.332 & -0.170 \\ 0 & 0 & 0 & 0 & 0.863 \end{pmatrix}$$

becomes dense

same, up to a permutation of coordinates

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 3 & 0 & 1 \\ 0 & 0 & 0 & 4 & 1 \\ 1 & 1 & 1 & 1 & 5 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1.414 & 0 & 0 & 0 \\ 0 & 0 & 1.732 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 1 & 0.707 & 0.577 & 0.5 & 1.708 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1.414 & 0 & 0 & 0.707 \\ 0 & 0 & 1.732 & 0 & 0.577 \\ 0 & 0 & 0 & 2 & 0.5 \\ 0 & 0 & 0 & 0 & 1.708 \end{pmatrix}$$

remains sparse

Question: When does a Cholesky decomposition keep the 0's?

Cholesky decomposition

Question: When does a Cholesky decomposition keep the 0's ?

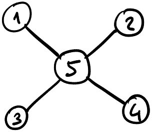
Theorem [Rose, Tarjan, Lueker, 1976]

Cholesky decomposition preserves the number of "zero entries"

if and only if

the graph with edge set $\{\{i,j\} \mid A_{i,j} \neq 0\}$ is chordal and $1, 2, \ldots, d$ is a perfect elimination order

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 3 & 0 & 1 \\ 0 & 0 & 0 & 4 & 1 \\ 1 & 1 & 1 & 1 & 5 \end{pmatrix}$$



NEED FOR RANDOM GENERATION

Chordal graphs are very studied:
People demand random generators!

NEED FOR RANDOM GENERATION

Chordal graphs are very studied:
People demand random generators!

At least 5 publications:

L. Markenzon, O. Vernet, and L. Araujo. Two methods for the generation of chordal graphs. *Annals of Operations Research*, 157(1):47–60, January 2008.

W. Sun and I. Bezáková. Sampling random chordal graphs by MCMC (student abstract). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:13929–13930, 04 2020.

A. Mukhopadhyay and M. Z. Rahman. Algorithms for generating strongly chordal graphs. In *Transactions on computational science XXXVIII*, volume 12620 of *Lecture Notes in Comput. Sci.*, pages 54–75. Springer, Berlin, 2021.

O. Şeker, P. Heggernes, T. Ekim, and Z. C. Taşkın. Generation of random chordal graphs using subtrees of a tree. *RAIRO Oper. Res.*, 56(2):565–582, 2022.

Ú. Hébert-Johnson, D. Lokshtanov, and E. Vigoda. Counting and sampling labeled chordal graphs in polynomial time. In *31st annual European Symposium on Algorithms*, volume 274 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 58, 17. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2023.

NEED FOR RANDOM GENERATION

Chordal graphs are very studied:
People demand random generators!

At least 5 publications:

Problems:

L. Markenzon, O. Vernet, and L. Araujo. Two methods for the generation of chordal graphs. *Annals of Operations Research*, 157(1):47–60, January 2008.

W. Sun and I. Bezáková. Sampling random chordal graphs by MCMC (student abstract). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:13929–13930, 04 2020.

A. Mukhopadhyay and M. Z. Rahman. Algorithms for generating strongly chordal graphs. In *Transactions on computational science XXXVIII*, volume 12620 of *Lecture Notes in Comput. Sci.*, pages 54–75. Springer, Berlin, 2021.

O. Şeker, P. Heggernes, T. Ekim, and Z. C. Taşkın. Generation of random chordal graphs using subtrees of a tree. *RAIRO Oper. Res.*, 56(2):565–582, 2022.

no control over probability distribution

Ú. Hébert-Johnson, D. Lokshtanov, and E. Vigoda. Counting and sampling labeled chordal graphs in polynomial time. In *31st annual European Symposium on Algorithms*, volume 274 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 58, 17. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2023.

NEED FOR RANDOM GENERATION

Chordal graphs are very studied:
People demand random generators!

At least 5 publications:

Problems:

L. Markenzon, O. Vernet, and L. Araujo. Two methods for the generation of chordal graphs. *Annals of Operations Research*, 157(1):47–60, January 2008.

W. Sun and I. Bezáková. Sampling random chordal graphs by MCMC (student abstract). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:13929–13930, 04 2020.

A. Mukhopadhyay and M. Z. Rahman. Algorithms for generating strongly chordal graphs. In *Transactions on computational science XXXVIII*, volume 12620 of *Lecture Notes in Comput. Sci.*, pages 54–75. Springer, Berlin, 2021.

O. Şeker, P. Heggernes, T. Ekim, and Z. C. Taşkın. Generation of random chordal graphs using subtrees of a tree. *RAIRO Oper. Res.*, 56(2):565–582, 2022.

no control over probability distribution

Ú. Hébert-Johnson, D. Lokshtanov, and E. Vigoda. Counting and sampling labeled chordal graphs in polynomial time. In *31st annual European Symposium on Algorithms*, volume 274 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 58, 17. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2023.

tractable up to 30 vertices

NEED FOR RANDOM GENERATION

Chordal graphs are very studied:
People demand random generators!

At least 5 publications:

Problems:

L. Markenzon, O. Vernet, and L. Araujo. Two methods for the generation of chordal graphs. *Annals of Operations Research*, 157(1):47–60, January 2008.

W. Sun and I. Bezáková. Sampling random chordal graphs by MCMC (student abstract). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:13929–13930, 04 2020.

A. Mukhopadhyay and M. Z. Rahman. Algorithms for generating strongly chordal graphs. In *Transactions on computational science XXXVIII*, volume 12620 of *Lecture Notes in Comput. Sci.*, pages 54–75. Springer, Berlin, 2021.

O. Şeker, P. Heggernes, T. Ekim, and Z. C. Taşkın. Generation of random chordal graphs using subtrees of a tree. *RAIRO Oper. Res.*, 56(2):565–582, 2022.

no control over probability distribution

Ú. Hébert-Johnson, D. Lokshtanov, and E. Vigoda. Counting and sampling labeled chordal graphs in polynomial time. In *31st annual European Symposium on Algorithms*, volume 274 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 58, 17. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2023.

tractable up to 30 vertices

Interesting Goal. Efficient uniform random sampler for chordal graphs

NEED FOR RANDOM GENERATION

Chordal graphs are very studied:
People demand random generators!

At least 5 publications:

Problems:

L. Markenzon, O. Vernet, and L. Araujo. Two methods for the generation of chordal graphs. *Annals of Operations Research*, 157(1):47–60, January 2008.

W. Sun and I. Bezáková. Sampling random chordal graphs by MCMC (student abstract). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:13929–13930, 04 2020.

A. Mukhopadhyay and M. Z. Rahman. Algorithms for generating strongly chordal graphs. In *Transactions on computational science XXXVIII*, volume 12620 of *Lecture Notes in Comput. Sci.*, pages 54–75. Springer, Berlin, 2021.

O. Şeker, P. Heggernes, T. Ekim, and Z. C. Taşkın. Generation of random chordal graphs using subtrees of a tree. *RAIRO Oper. Res.*, 56(2):565–582, 2022.

no control over probability distribution

Ú. Hébert-Johnson, D. Lokshtanov, and E. Vigoda. Counting and sampling labeled chordal graphs in polynomial time. In *31st annual European Symposium on Algorithms*, volume 274 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 58, 17. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2023.

tractable up to 30 vertices

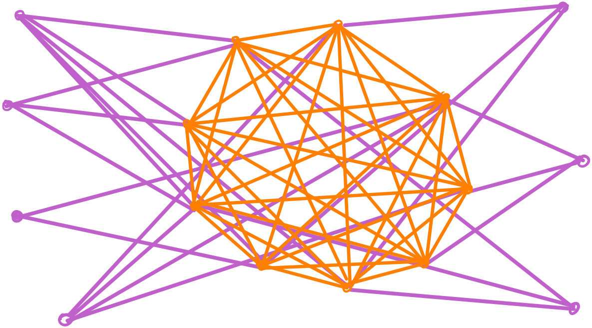
Interesting Goal? Efficient uniform random sampler for chordal graphs

That would be interesting, right?

UNIFORM DISTRIBUTION IS LAME

All chordal graphs look alike under uniform distribution: about 1/2 the vertices form a clique, the rest form an independent set

[Bender Richmond Wormald 1985]



AN INTERESTING PARAMETER: TREEWIDTH

treewidth = graph parameter of paramount importance
notably in parametrized complexity
[cf. Courcelle's theorem]

AN INTERESTING PARAMETER: TREEWIDTH

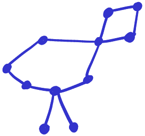
treewidth = graph parameter of paramount importance notably in parametrized complexity

[cf. Courcelle's theorem]

DEFINITION

$$\text{treewidth}(G) = \min_{\substack{\text{chordal graph } C \\ \text{such that } G \subseteq C}} \text{size of the largest clique in } C - 1$$

Ex



treewidth =

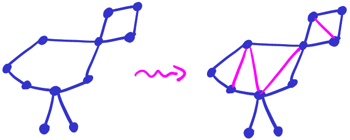
AN INTERESTING PARAMETER: TREEWIDTH

treewidth = graph parameter of paramount importance notably in parametrized complexity
[cf. Courcelle's theorem]

DEFINITION

$$\text{treewidth}(G) = \min_{\substack{\text{chordal graph } C \\ \text{such that } G \subseteq C}} \text{size of the largest clique in } C - 1$$

Ex



treewidth = 2 max clique = 3

AN INTERESTING PARAMETER: TREEWIDTH

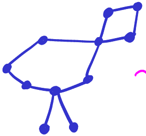
treewidth = graph parameter of paramount importance notably in parametrized complexity

[cf. Courcelle's theorem]

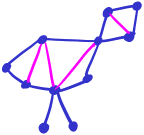
DEFINITION

$$\text{treewidth}(G) = \min_{\substack{\text{chordal graph } C \\ \text{such that } G \subseteq C}} \text{size of the largest clique in } C - 1$$

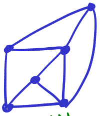
Ex



treewidth = 2



max clique = 3



treewidth =

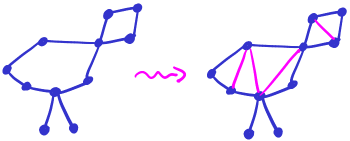
AN INTERESTING PARAMETER: TREEWIDTH

treewidth = graph parameter of paramount importance notably in parametrized complexity [cf. Courcelle's theorem]

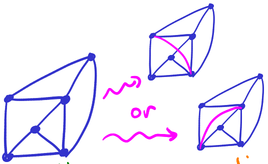
DEFINITION

$$\text{treewidth}(G) = \min_{\substack{\text{chordal graph } C \\ \text{such that } G \subseteq C}} \text{size of the largest clique in } C \; - 1$$

Ex



treewidth = 2 max clique = 3



treewidth = 3 max clique = 4

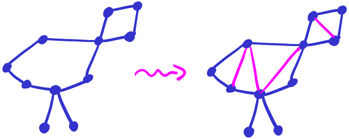
AN INTERESTING PARAMETER: TREEWIDTH

treewidth = graph parameter of paramount importance notably in parametrized complexity [cf. Courcelle's theorem]

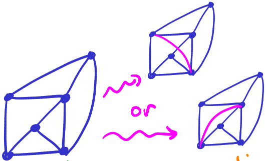
DEFINITION

$$\text{treewidth}(G) = \min_{\substack{\text{chordal graph } C \\ \text{such that } G \subseteq C}} \text{size of the largest clique in } C - 1$$

Ex



treewidth = 2   max clique = 3



treewidth = 3   max clique = 4

<u>Heuristic</u> Real graphs have small treewidth.

MORAL OF THIS INTRODUCTION

Chordal graphs are important

Perfect elimination ordering matters

Small treewidth is better

MORAL OF THIS INTRODUCTION

Chordal graphs are important

Perfect elimination ordering matters

Small treewidth is better

What we wish to study: chordal graphs with treewidth $\leq k$ with a perfect elimination ordering

MORAL OF THIS INTRODUCTION

Chordal graphs are important

Perfect elimination ordering matters

Small treewidth is better

__What we wish to study__: chordal graphs with treewidth $\leq k$ with a perfect elimination ordering

__Note__: chordal graphs with treewidth $\leq k$ were studied by [Castellví et al.]

PART II

WHAT WE DID

(TREEWIDTH ≤ 2)

INCREASING 1,2-TREES

DEFINITION

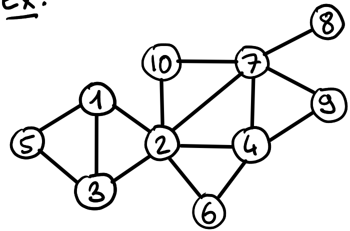
increasing 1,2-tree =
- start from 1
- pick vertex → attach leaf

or

pick edge → attach triangle

- repeat

Ex:



increasing 1,2-tree = chordal graph with treewidth ≤ 2 with a perfect elimination ordering

INCREASING 1,2 - TREES

DEFINITION

increasing 1,2 - tree = • start from 1

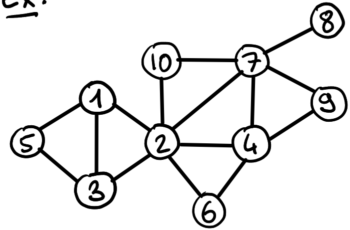
• pick vertex → attach leaf 

or

pick edge → attach triangle 

• repeat

Ex:



Objectives

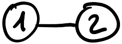
• Count the increasing 1,2-trees with n vertices

• Sample a random increasing 1,2-tree with n vertices uniformly

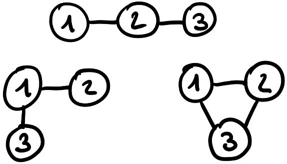
INCREASING 1,2 - TREES

$n = 1$

$n = 2$



$n = 3$



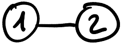
Objectives

- Count the increasing 1,2-trees with n vertices
- Sample a random increasing 1,2-tree with n vertices uniformly

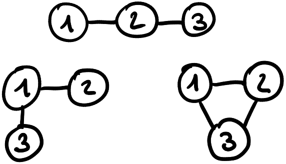
INCREASING 1,2-TREES

$n=1$

$n=2$



$n=3$



Anyone willing to do $n=4$?

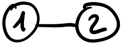
Objectives

- Count the increasing 1,2-trees with n vertices
- Sample a random increasing 1,2-tree with n vertices uniformly

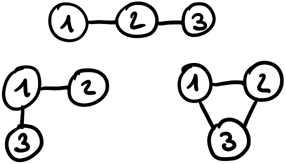
INCREASING 1,2-TREES

$n=1$

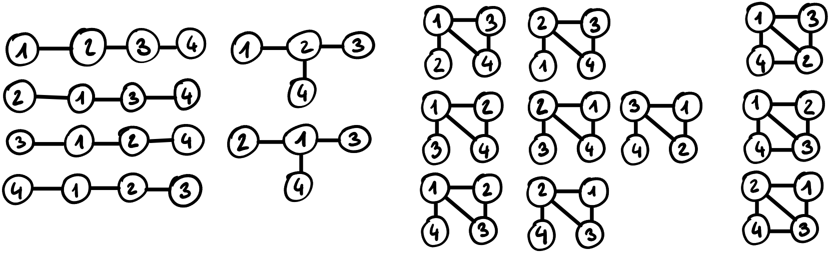
$n=2$



$n=3$



$n=4$



RECURRENCE FOR INCREASING 1,2-TREES

$c_{n,m}$ = number of increasing 1,2-trees with n vertices and m edges

$$\boxed{c_{n,m} = (n-1)\, c_{n-1,m-1} + (m-2)\, c_{n-1,m-2}}$$

RECURRENCE FOR INCREASING 1,2 -TREES

$c_{n,m}$ = number of increasing 1,2-trees with n vertices and m edges

$$\boxed{c_{n,m} = (n-1)\, c_{n-1,m-1} + (m-2)\, c_{n-1,m-2}}$$

Size	1	2	3	4	5	6
Number of increasing 1,2-trees	1	1	3	16	125	1296

RECURRENCE FOR INCREASING 1,2-TREES

$c_{n,m}$ = number of increasing 1,2-trees with n vertices and m edges

$$\boxed{c_{n,m} = (n-1)\, c_{n-1,m-1} + (m-2)\, c_{n-1,m-2}}$$

Size	1	2	3	4	5	6
Number of increasing 1,2-trees	1	1	3	16	125	1296

These numbers seem to be n^{n-2}, the number of Cayley trees?

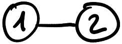
CAYLEY TREES

Cayley tree = labelled tree

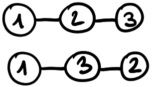
> [Cayley 1889] Cayley trees with n vertices are counted by n^{n-2}.

$n=1$

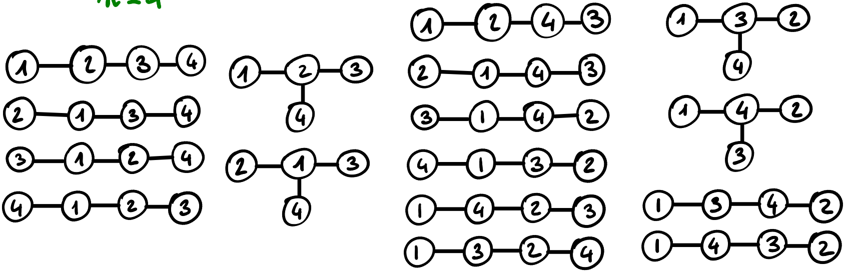
$n=2$



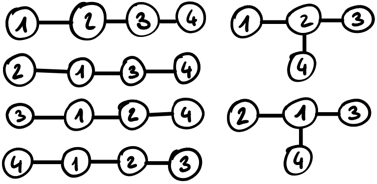
$n=3$

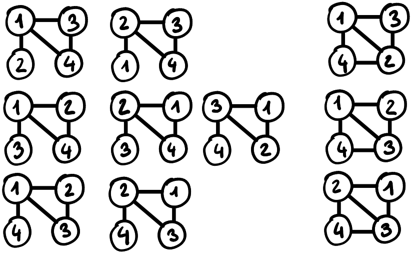


$n=4$



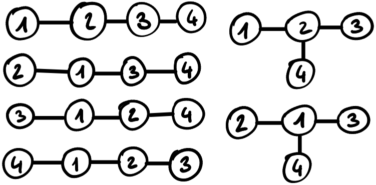
CAYLEY TREES & INCREASING 1,2-TREES

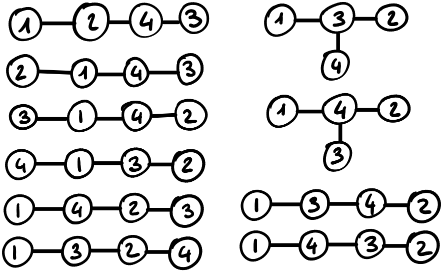




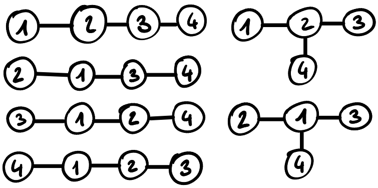
Increasing 1,2-trees of size 4

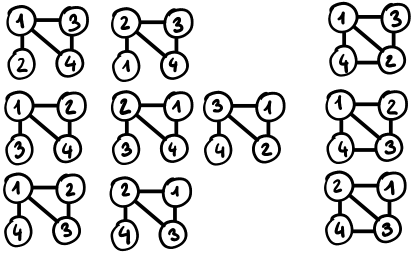
Cayley trees of size 4





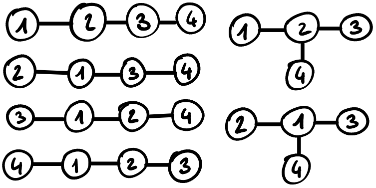
Number of increasing 1,2-trees = number of Cayley trees ?

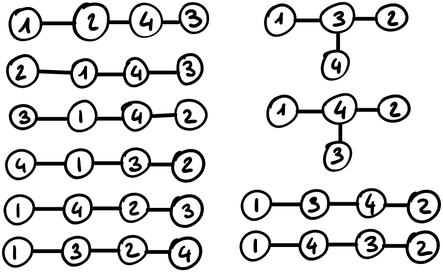




Increasing 1,2-trees of size 4

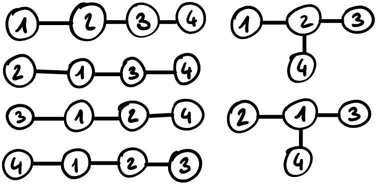
Cayley trees of size 4

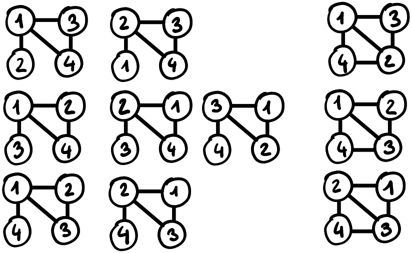




Number of increasing 1,2-trees
= number of Cayley trees ?

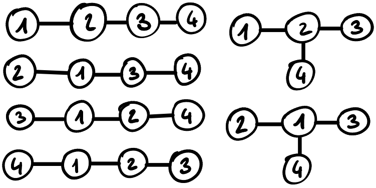
"It's a matter of 10 minutes."
Matthieu Dien

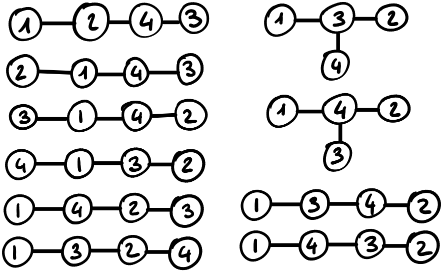


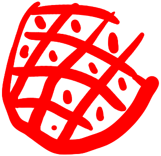


Increasing 1,2-trees of size 4

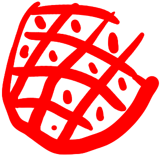
Cayley trees of size 4







A FEW YEARS LATER

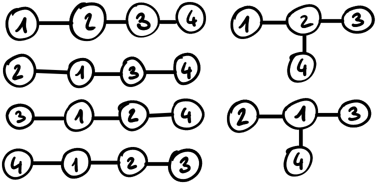


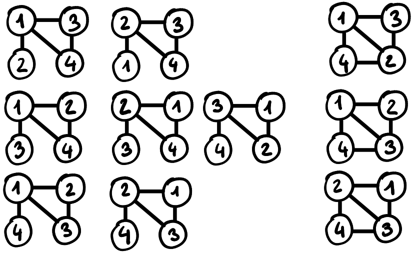
A FEW YEARS LATER

Theorem [Courtiel Dien Dorbec]

number of increasing 1,2-trees with n vertices

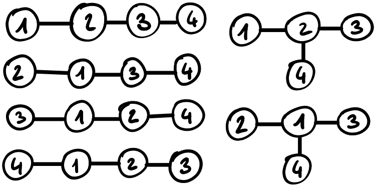
= number of Cayley trees with n vertices triangles ⟷ ???

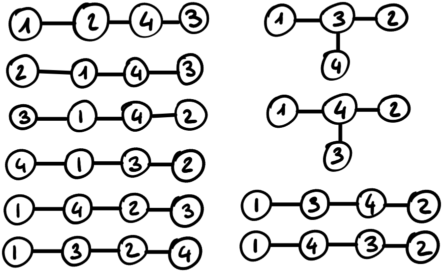




Increasing 1,2-trees of size 4

Cayley trees of size 4

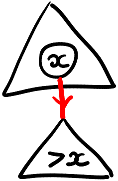




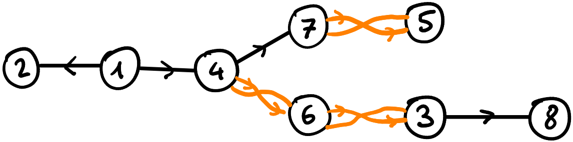
TWISTS

Root at 1

increasing edge = edge from a vertex x to a subtree where all vertices have label $> x$



twist = non-increasing edge.

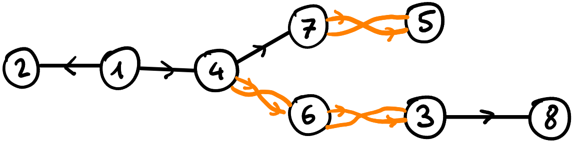


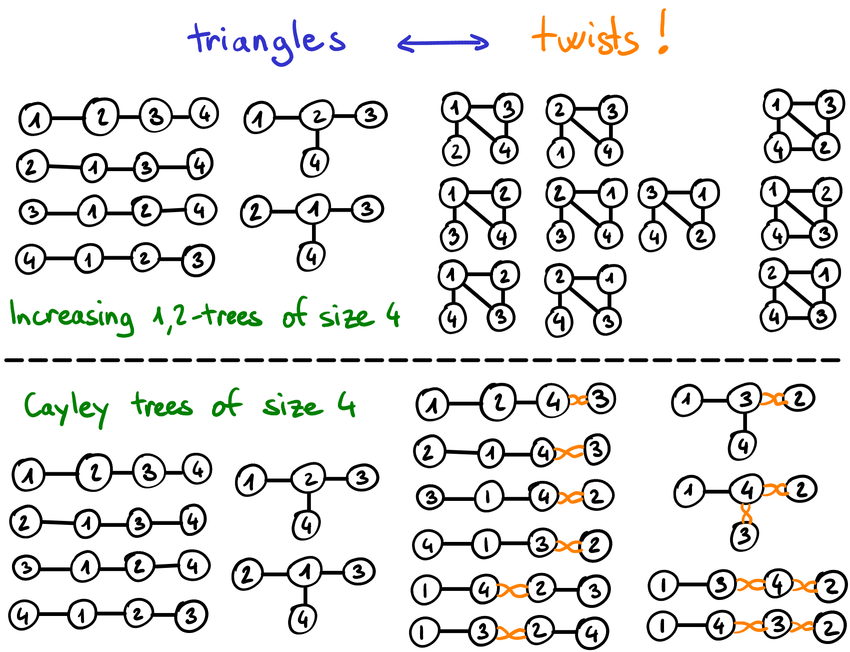
OUR THEOREM

Theorem [Courtiel Dien Dorbec]

number of increasing 1,2-trees with n vertices and m triangles

= number of Cayley trees with n vertices and m twists




triangles ⟷ twists !
Increasing 1,2-trees of size 4
Cayley trees of size 4

OUR THEOREM

Theorem [Courtiel Dien Dorbec]

number of increasing 1,2-trees with n vertices and m triangles

= number of Cayley trees with n vertices and m twists

$\boxed{\text{3 proofs}}$

1. Recurrence [Shor 1995]
2. Analytic (Solving a PDE)
3. Bijection

OUR THEOREM

Theorem [Courtiel Dien Dorbec]

number of increasing 1,2-trees with n vertices and m triangles

= number of Cayley trees with n vertices and m twists

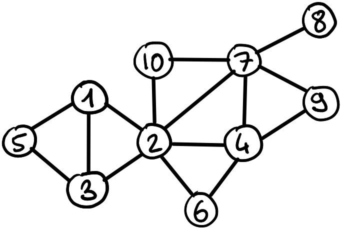
3 proofs

1. Recurrence [Shor 1995]

2. Analytic (Solving a PDE)

3. Bijection

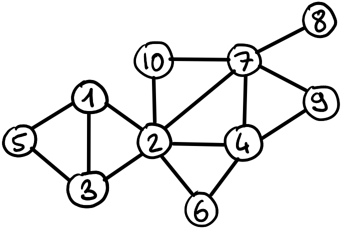
BIJECTION



in a reversible way

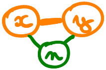
Cayley tree

BIJECTION

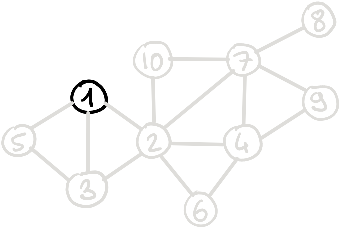


Step 1. Arrange the edges into a plane forest

Rule 

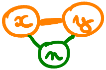
Rule 

BIJECTION

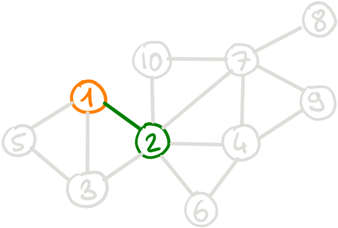


Step 1. Arrange the edges into a plane forest

Rule 

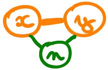
Rule 

BIJECTION

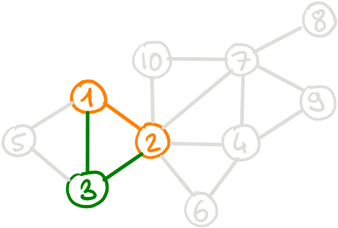


Step 1. Arrange the edges into a plane forest

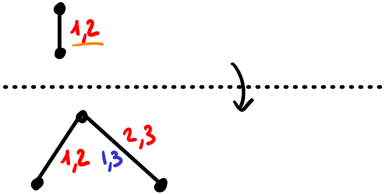
Rule 

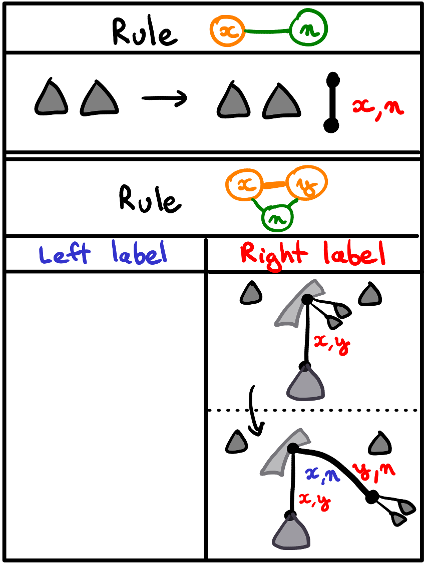
Rule 

BIJECTION

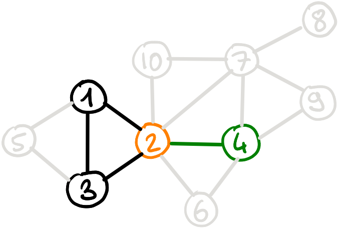


Step 1. Arrange the edges into a plane forest

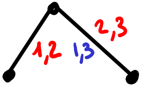


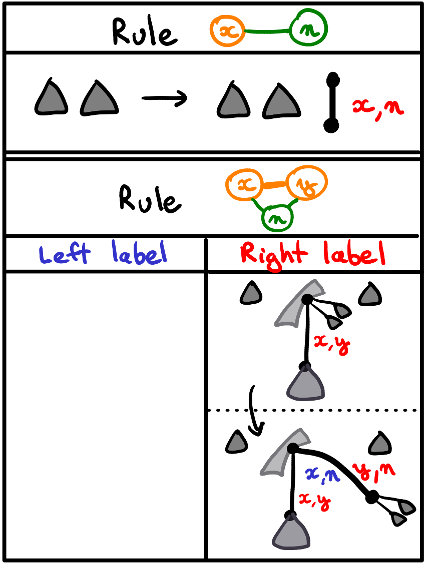


BIJECTION

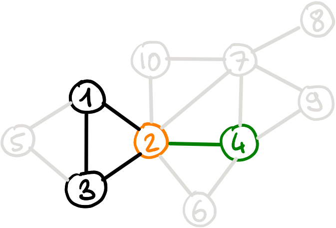


Step 1. Arrange the edges into a plane forest

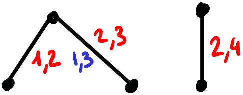


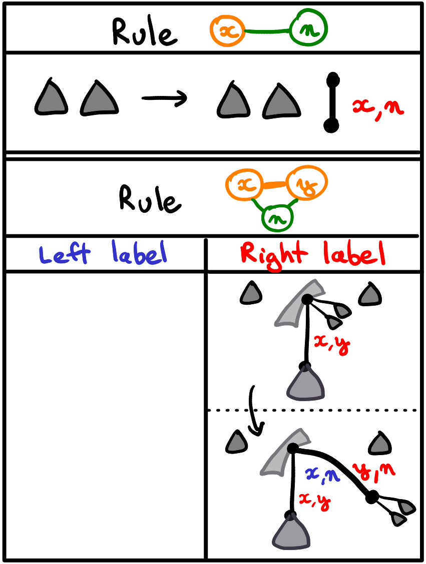


BIJECTION

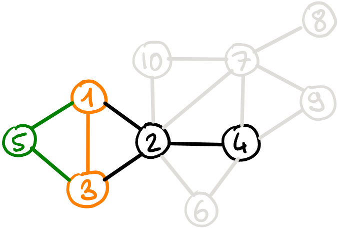


Step 1. Arrange the edges into a plane forest

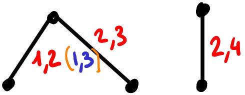


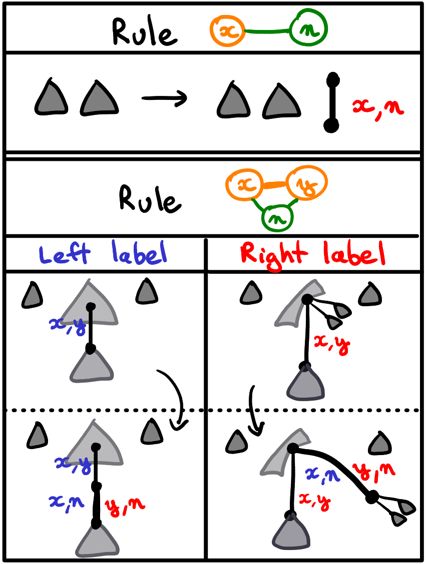


BIJECTION

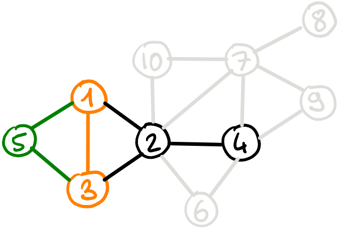


Step 1. Arrange the edges into a plane forest

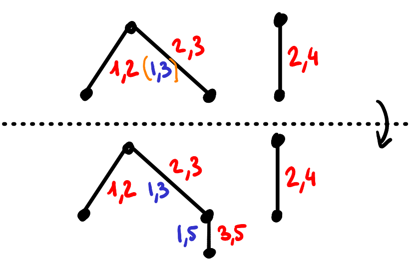


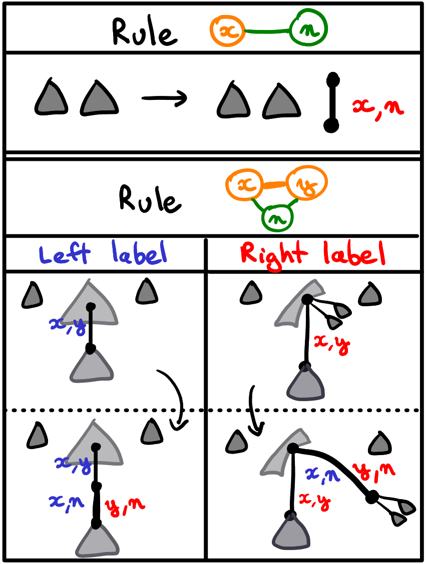


BIJECTION

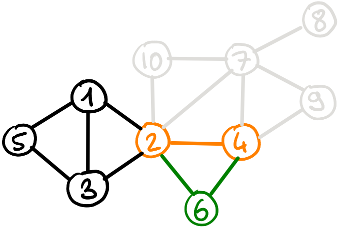


Step 1. Arrange the edges into a plane forest

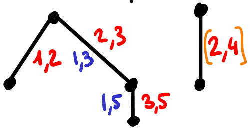


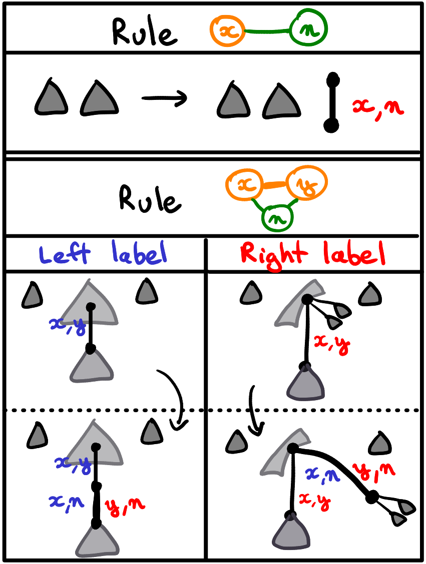


BIJECTION

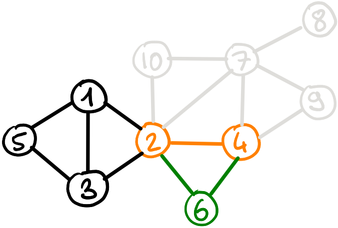


Step 1. Arrange the edges into a plane forest

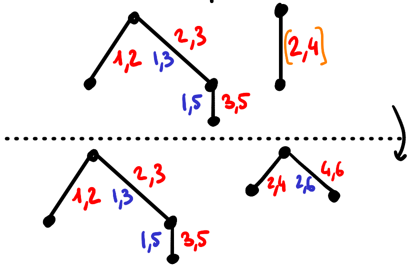


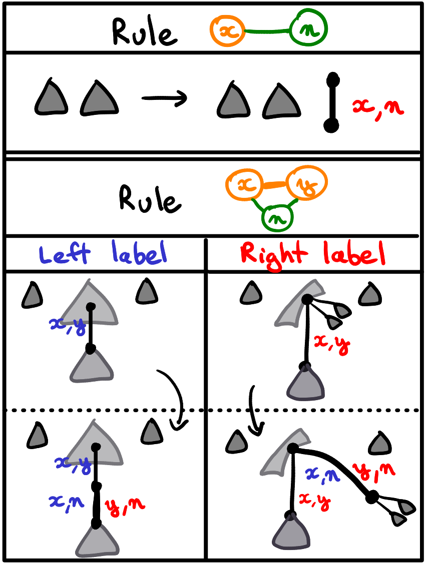


BIJECTION

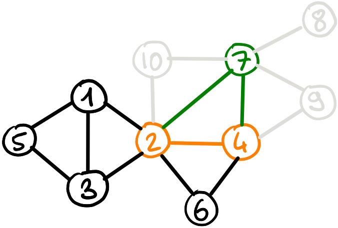


Step 1. Arrange the edges into a plane forest

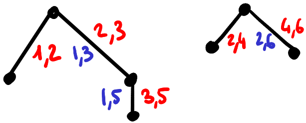


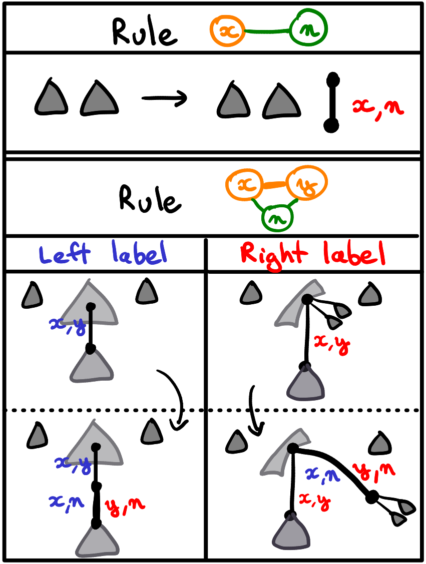


BIJECTION

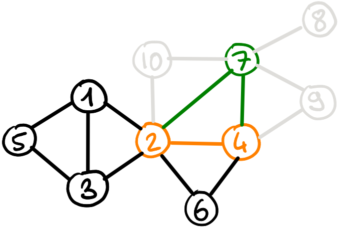


Step 1. Arrange the edges into a plane forest

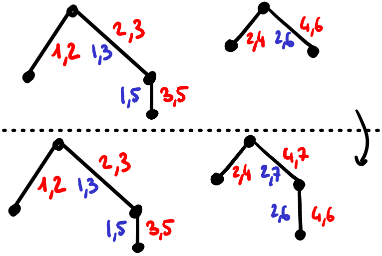


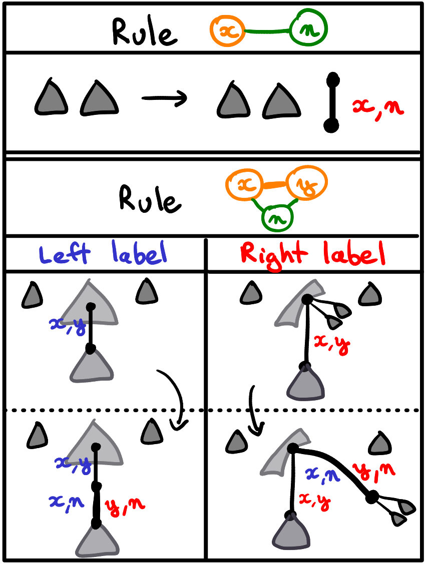


BIJECTION

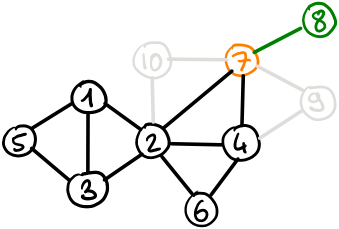


Step 1. Arrange the edges into a plane forest

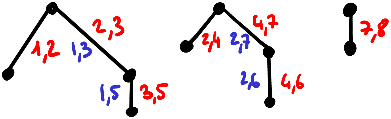


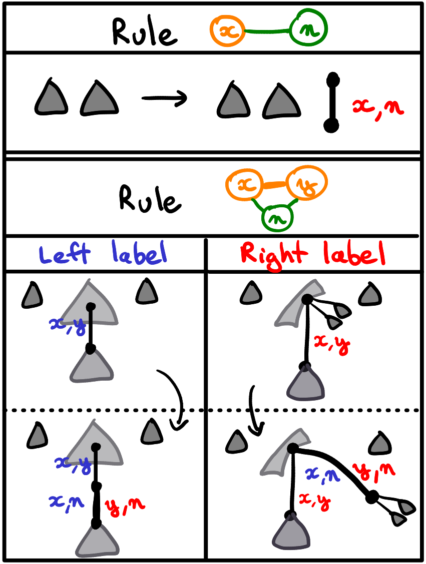


BIJECTION

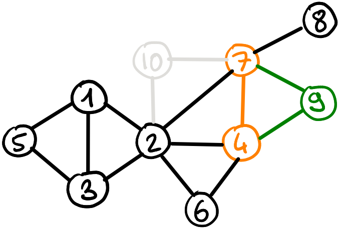


Step 1. Arrange the edges into a plane forest

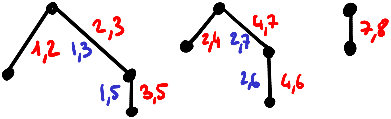


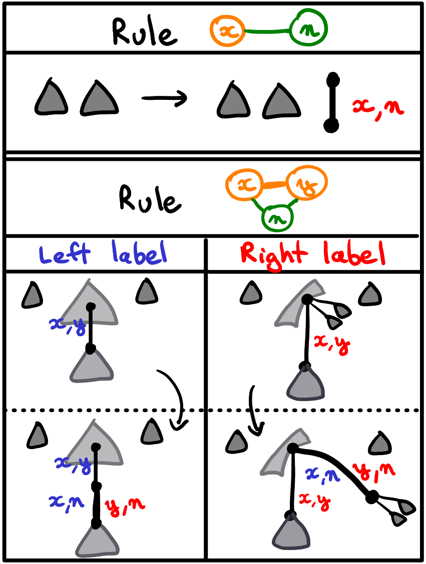


BIJECTION

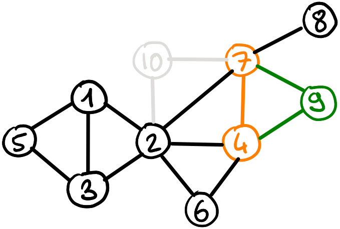


Step 1. Arrange the edges into a plane forest

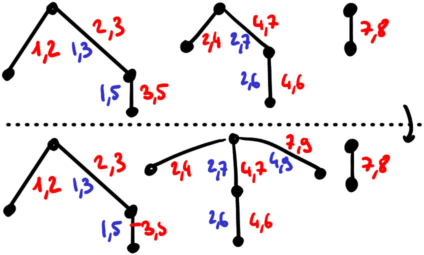


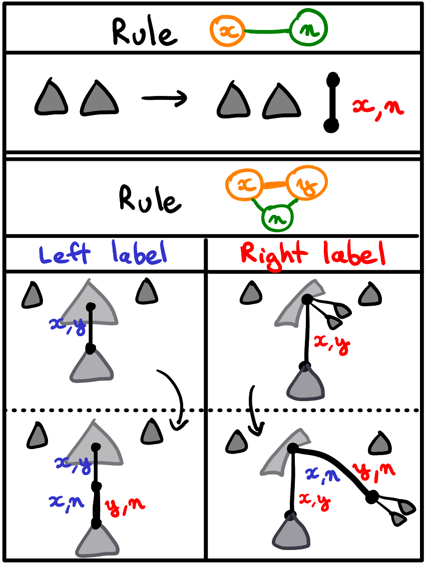


BIJECTION

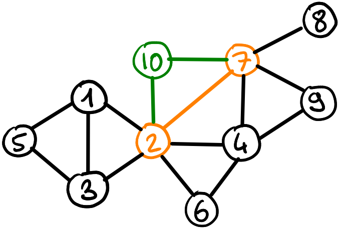


Step 1. Arrange the edges into a plane forest

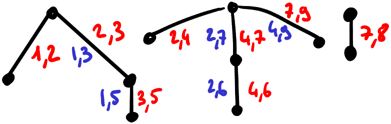


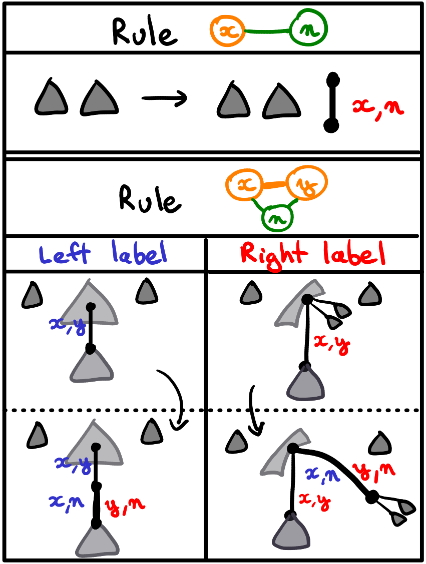


BIJECTION

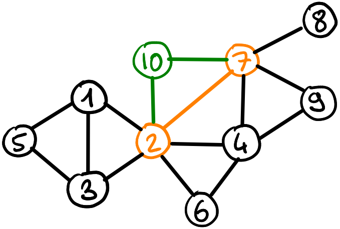


Step 1. Arrange the edges into a plane forest

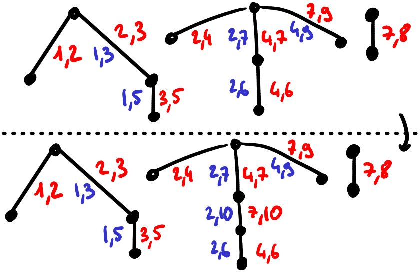


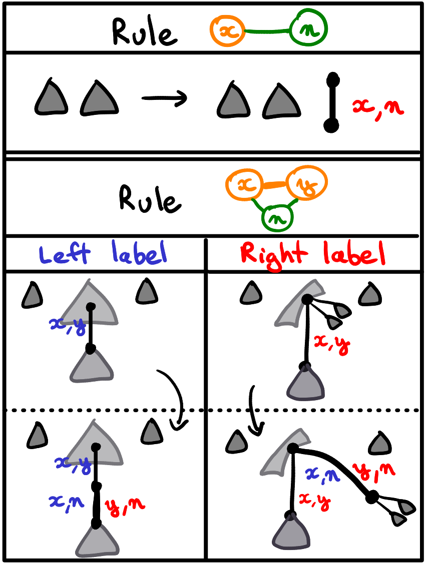


BIJECTION

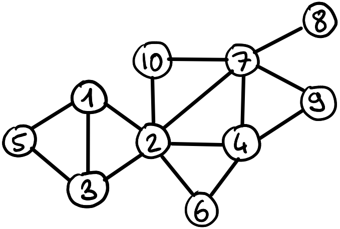


Step 1. Arrange the edges into a plane forest

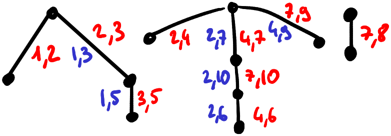


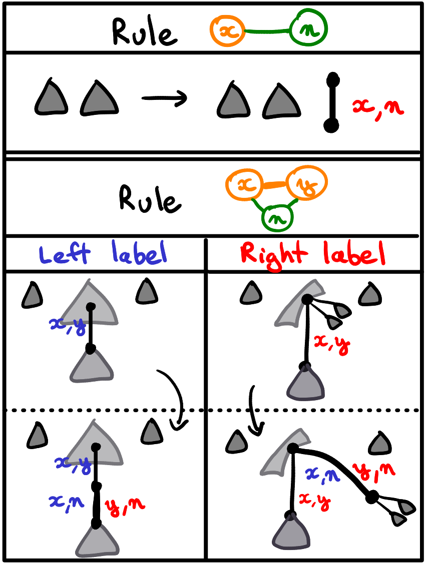


BIJECTION

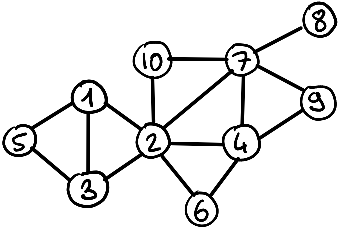


Step 1. Arrange the edges into a plane forest

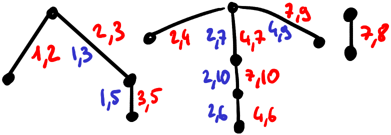




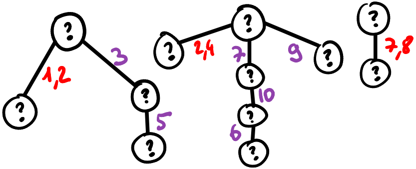
BIJECTION



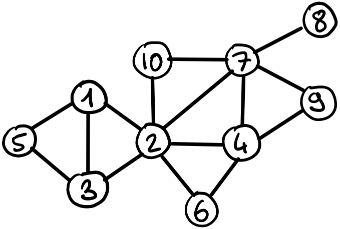
Step 1. Arrange the edges into a plane forest



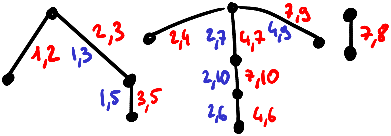
Step 2. Label the vertices



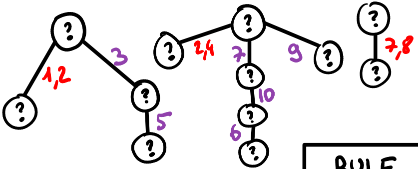
BIJECTION



Step 1. Arrange the edges into a plane forest



Step 2. Label the vertices

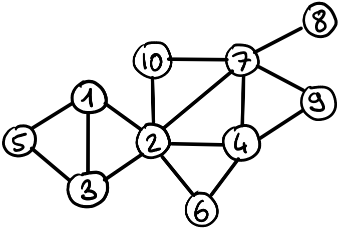


RULE

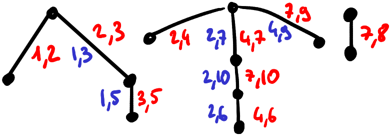
MAYOR
MATCHMAKER
HIKER

Rightmost DFS:
match the i^{th} vertex
with the i^{th} edge

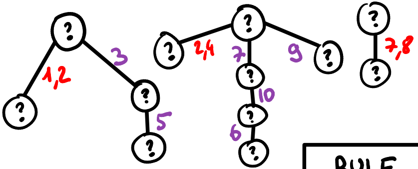
BIJECTION



Step 1. Arrange the edges into a plane forest



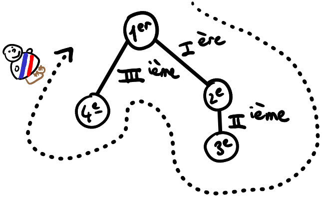
Step 2. Label the vertices



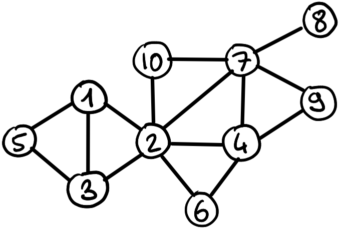
RULE

MAYOR MATCHMAKER HIKER

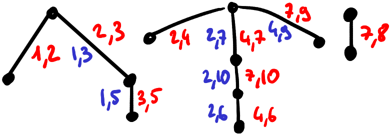
Rightmost DFS: match the i^{th} vertex with the i^{th} edge



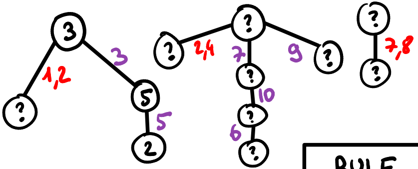
BIJECTION



Step 1. Arrange the edges into a plane forest



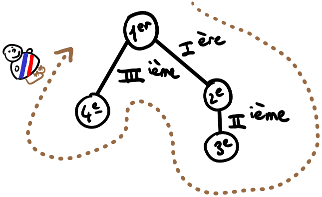
Step 2. Label the vertices



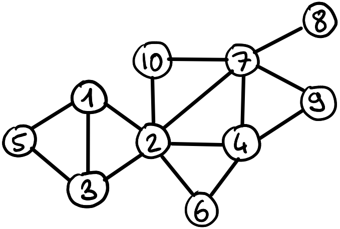
RULE

MAYOR MATCHMAKER HIKER

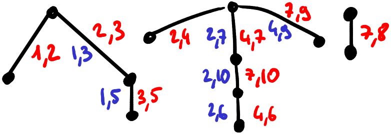
Rightmost DFS: match the i^{th} vertex with the i^{th} edge



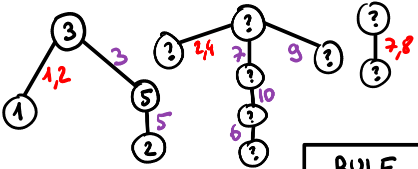
BIJECTION



Step 1. Arrange the edges into a plane forest



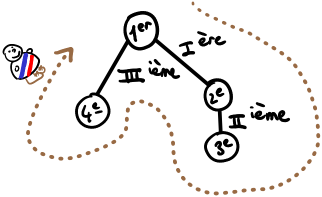
Step 2. Label the vertices



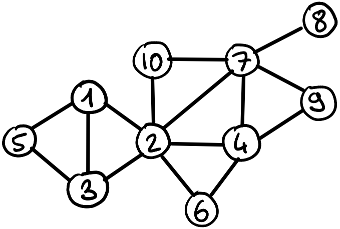
RULE

MAYOR MATCHMAKER HIKER

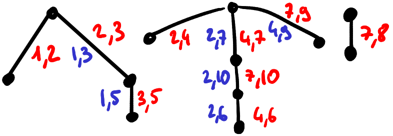
Rightmost DFS:
match the
i^{th} vertex
with the i^{th} edge



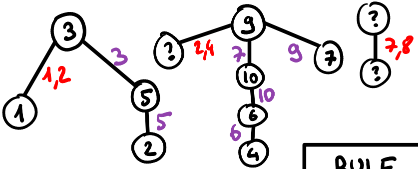
BIJECTION



Step 1. Arrange the edges into a plane forest



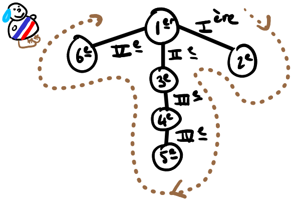
Step 2. Label the vertices



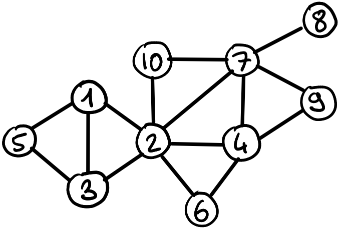
RULE

MAYOR MATCHMAKER HIKER

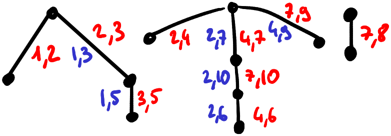
Rightmost DFS: match the i^{th} vertex with the i^{th} edge



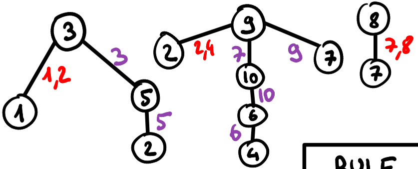
BIJECTION



Step 1. Arrange the edges into a plane forest



Step 2. Label the vertices

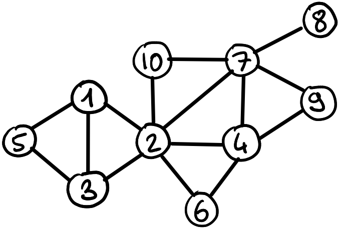


RULE

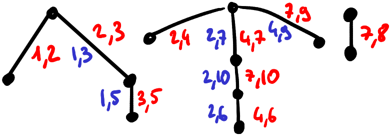
MAYOR
MATCHMAKER
HIKER

Rightmost DFS:
match the i^{th} vertex
with the i^{th} edge

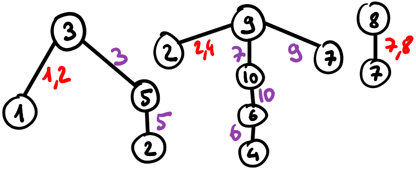
BIJECTION



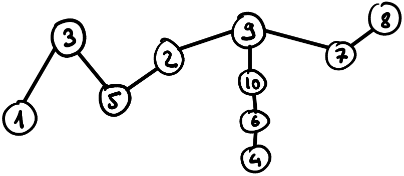
Step 1. Arrange the edges into a plane forest



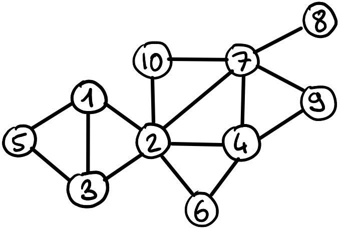
Step 2. Label the vertices



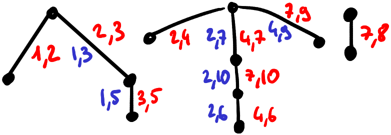
Step 3. Merge vertices with same label & forget the rest



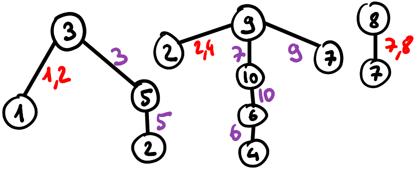
BIJECTION



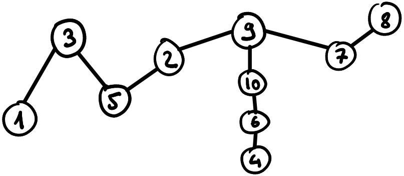
Step 1. Arrange the edges into a plane forest



Step 2. Label the vertices

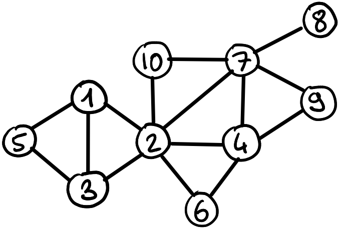


Step 3. Merge vertices with same label & forget the rest

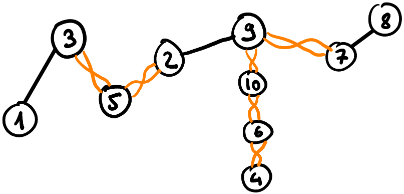


Theorem: That's a bijection!

BIJECTION



INCREASING 1,2-TREE



CAYLEY TREE

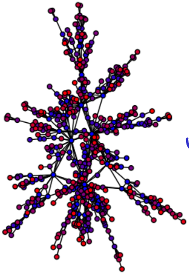
This bijection sends:

- vertices to vertices
- triangles to twists
- increasing trees to the same increasing trees
- increasing 2-trees to Cayley trees with only twists (except 1 edge)

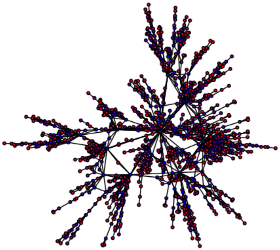
APPLICATION: RANDOM GENERATION

Algorithm sampling a random increasing 1,2-tree with n vertices in linear time:

$$\boxed{\text{Sample uniform Cayley tree}} \longrightarrow \boxed{\text{Reverse bijection}} \longrightarrow \boxed{\text{Profit}}$$



500 vertices



1200 vertices

A LAST SLIDE FOR THE EXPERTS

What about the analytic proof?

A LAST SLIDE FOR THE EXPERTS

What about the analytic proof?

The exponential generating function $C(z,u)$ of increasing 1,2-trees satisfies the PDE

counts vertices

counts edges

$$\frac{\partial C}{\partial z} = 1 + zu\frac{\partial C}{\partial z} + u^3\frac{\partial C}{\partial u}$$

with initial condition $C(0,u)=0$

The unique solution is

$$C(z,u) = \mathrm{Cayley}\left(\frac{uz+u-1}{u}e^{\frac{1-u}{u}}\right) + \frac{1}{2u^2} - \frac{1}{2}$$

where

$$\mathrm{Cayley}(x) = \sum_{n\geq 0} \frac{n^{n-2}}{n!}x^n .$$

A LAST SLIDE FOR THE EXPERTS

What about the analytic proof?

The exponential generating function $C(z, u)$ of increasing 1,2-trees satisfies the PDE

counts vertices

counts edges

$$\frac{\partial C}{\partial z} = 1 + zu\frac{\partial C}{\partial z} + u^3\frac{\partial C}{\partial u}$$ with initial condition $C(0, u) = 0$

The unique solution is

$$C(z,u) = \mathrm{Cayley}\left(\frac{uz+u-1}{u}e^{\frac{1-u}{u}}\right) + \frac{1}{2u^2} - \frac{1}{2}$$

where

$$\mathrm{Cayley}(x) = \sum_{n\geq 0} \frac{n^{n-2}}{n!}x^n.$$

<u>Open question</u>: Is there a combinatorial interpretation?

PERSPECTIVES

We solved: chordal graphs with treewidth $\leq k$ with a perfect elimination ordering for $k = 2$.

PERSPECTIVES

We solved: chordal graphs with treewidth $\leq k$ with a perfect elimination ordering for $k = 2$.

$\boxed{\text{What about other } k\text{'s?}}$

PERSPECTIVES

We solved: chordal graphs with treewidth $\leq k$ with a perfect elimination ordering for $k = 2$.

$$\boxed{\text{What about other } k\text{'s?}}$$

Comparison with other models:

- uniform chordal graphs with treewidth $\leq k$ [Castellví et al]
- chordal graphs with treewidth $\leq k$ with a perfect elimination ordering seen as Pólya urn model.

MERCI !

ET VIENS DANSER
LE TWIST !