Cayley's Formula (cont'd).

[Borchard, Cayley, Sylvester]

$$\boxed{\#\text{ trees on } n \text{ labelled vertices} = n^{n-2}}$$

2 bijective proofs:

Proof # 1 [Eğecioğlu - Remmel '1986]

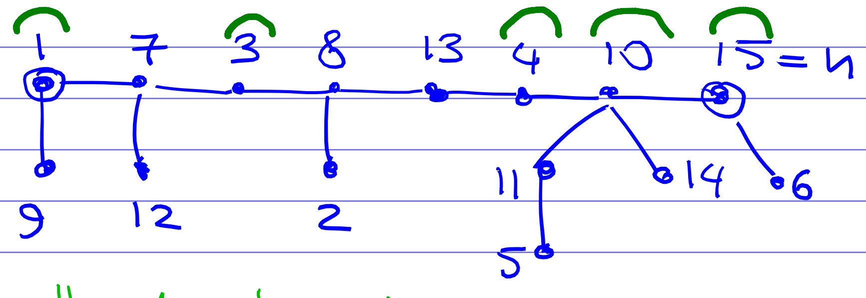
$$\left\{\begin{array}{l}\text{trees } T \text{ on} \\ n \text{ vertices}\end{array}\right\} \overset{\text{bij}}{\longleftrightarrow} \left\{\begin{array}{l}\text{maps } f: [n] \to [n] \\ f(1)=1,\ f(n)=n\end{array}\right\}$$

Example

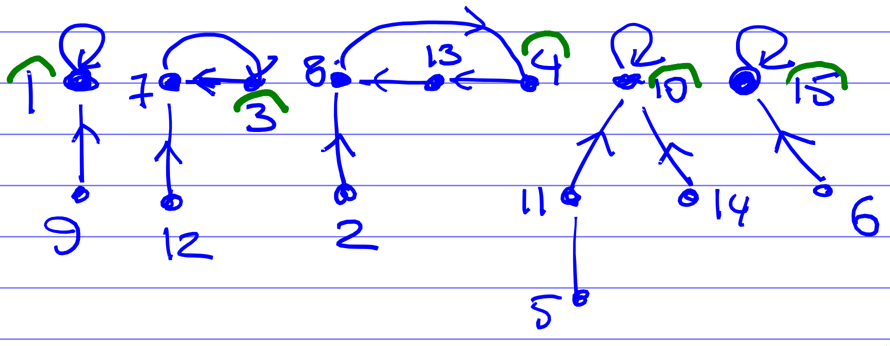
mark all vertices on the path from vertex 1 to vertex n which are less than all following vertices

a tree

$T =$



direct all edges towards vertex 1 & replace the path from 1 to n by the collection of cycles, as shown:

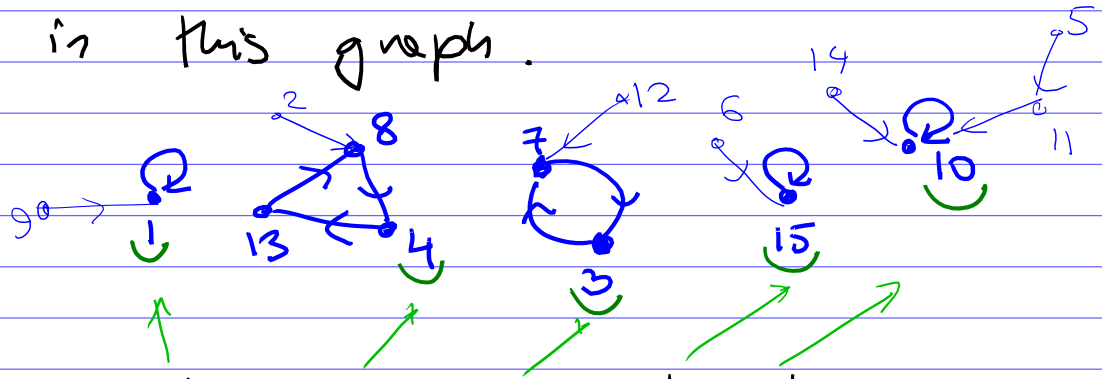


this directed graph represents the following map $f: [n] \to [n]$:

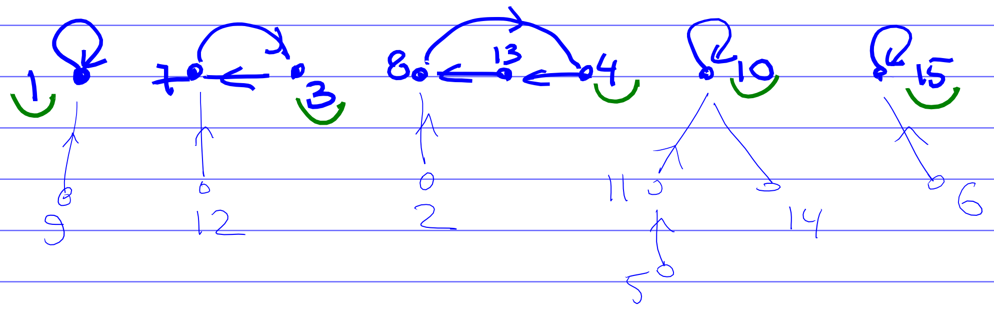
$$f: \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 1 & 8 & 7 & 13 & 11 & 15 & 3 & 4 & 1 & 10 & 10 & 7 & 8 & 10 & 15 \end{pmatrix}$$

The inverse construction $f \rightsquigarrow T$:

- draw a map f as the directed graph with edges $i \to f(i)$
- consider all directed cycles in this graph.

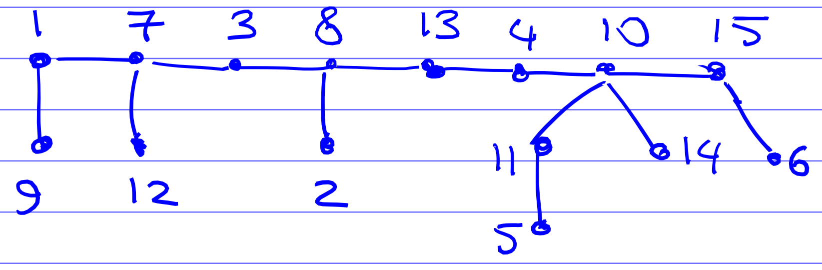


- mark the minimal elements in the cycles
- arrange the cycles so that their minimal elements are increase from left to right, and are in the end of cycles, as shown:



- Replace the cycles by the path from 1 to n and erase the arrows on the edges:

$T =$



So the map $T \leftrightarrow f$ is a bijection.

Proof #2 [Prüfer, 1918]

Ernst Paul Heinz Prüfer (1896 – 1934)

Prüfer code of a tree

$T \rightsquigarrow (c_1, c_2, \dots, c_{n-2})$

$c_i \in \{1, 2, \dots, n\}, \ \forall i$

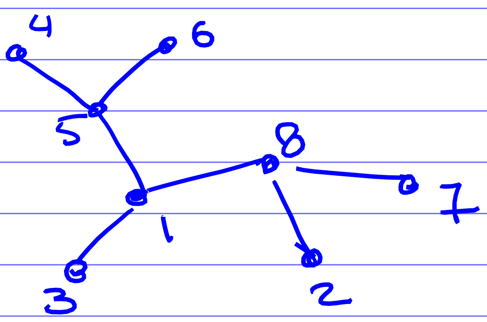
Coding:

- Find the leaf (i.e. vertex of degree 1) with minimal label ℓ
- Let (ℓ, c) be the (unique) edge incident to ℓ.
- Record the label c.
- Erase the edge (ℓ, c).

Repeat these steps $n-2$ times.

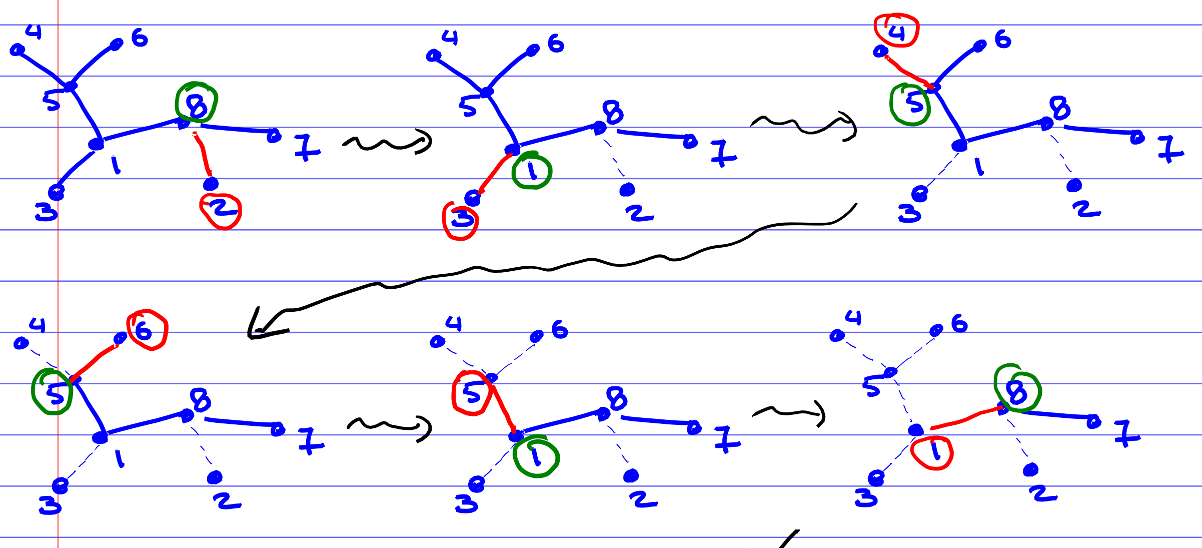
Example

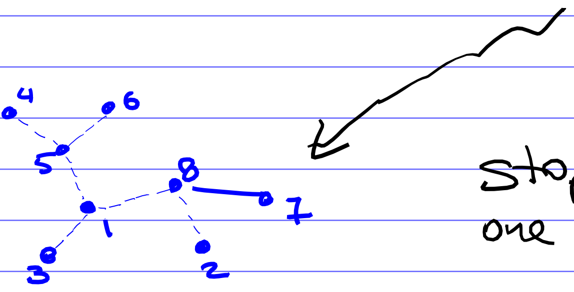
$T =$



min leaf	2	3	4	6	5	1
attached to	8	1	5	5	1	8

this is Prüfer code





stop when only one edge remains

Theorem [Prüfer]

This construction

$$T \mapsto \operatorname{code}(T) = (c_1, \dots, c_{n-2})$$

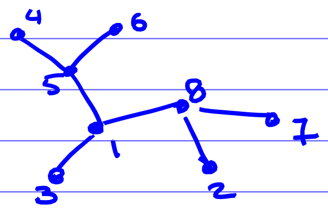
gives a bijection between all labelled trees T on n vertices and all sequences $(c_1, \dots, c_{n-2})$ with $c_i \in \{1, 2, \dots, n\}$.

Proof. We need to construct the inverse map, (i.e. decoding)

$$(c_1, c_2, \dots, c_{n-2}) \mapsto T.$$

Observation. For $i \in \{1, \dots, n\}$, the label i appears in code(T) exactly $\deg_T(i) - 1$ times.

Example

$T =$ 

$\mapsto \operatorname{code}(T) = (8, 1, 5, 5, 1, 8)$

$\deg(1) = 3$, $\deg(2) = 1$, ...

In particular, the labels of the leaves in T are exactly all $i \in \{1, \dots, n\}$ that do not appear in code(T).

Proof of the observation:

We record i in code(T) every time when we remove an edge incident to i, except the last edge incident to i (when i acts as a leaf)

This observation allows us to construct the decoding procedure.

<u>Decoding</u> : $(c_1, \dots, c_{n-2}) \mapsto T$.

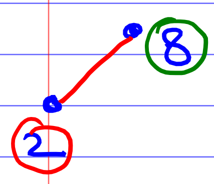
- Initially set
  $C = (c_1, \dots, c_{n-2})$ and
  $L = \{1, 2, \dots, n\}$

- Find the minimal element ℓ of L that <u>does not</u> appear in C.
  Let c be the first entry in C.
- Connect ℓ & c by an edge (ℓ, c).
- Remove ℓ from L, and remove c from C.

Repeat these steps $n-2$ times (untill C becomes empty)

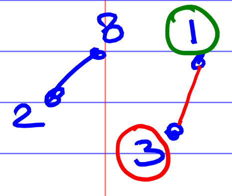
- Then connect the remaining two elements of L by an edge, and stop.

Example. $n=8$, $(c_1,\dots,c_6)=(8,1,5,5,1,8)$



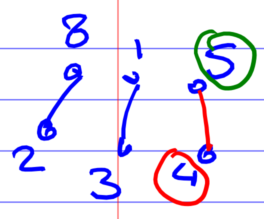
$C = (8,1,5,5,1,8)$

$L = \{1,2,3,4,5,6,7,8\}$



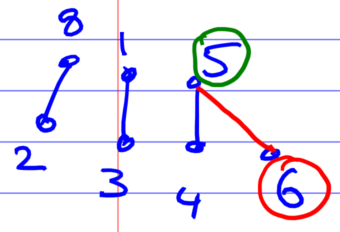
$(1,5,5,1,8)$

$\{1,3,4,5,6,7,8\}$



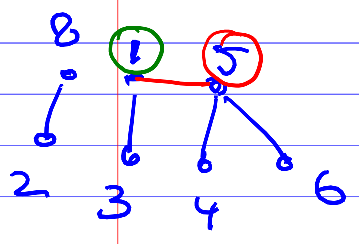
$(5,5,1,8)$

$\{1,4,5,6,7,8\}$



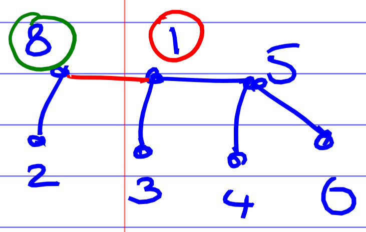
$(5,1,8)$

$\{1,5,6,7,8\}$



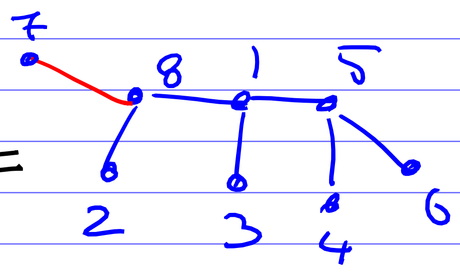
$(1,8)$

$\{1,5,7,8\}$



(8)

$\{1,7,8\}$

$T =$ 

$\emptyset$

$\{7,8\}$

Using the observation, we easily see that the decoding procedure is inverse to the coding procedre. (The edges are added in the same order as they were removed during coding).

So $T \longrightarrow \mathrm{code}(T)$ is a bijection.

Moreover, the observation implies.

Theorem $T \mapsto \mathrm{code}(T)$ is a bijection between all labelled trees with given degrees $d_1, d_2, \dots, d_n$ of vertices and sequences $(c_1, \dots, c_n)$ such that $\forall i \in [n]$, $\#\{j : c_j = i\} = d_i - 1$.

So we've got a bijective proof of the formula

$$\#\left\{\begin{array}{l}\text{labelled trees} \\ \text{with } n \text{ vertices} \\ \text{of degrees} \\ d_1, d_2, \dots, d_n\end{array}\right\} = \binom{n-2}{d_1-1 \;\; d_2-1 \;\; \dots \;\; d_n-1}$$

the multinomial coefficient $= \dfrac{(n-2)!}{(d_1-1)! \dots (d_n-1)!}$

Example. $(d_1, \dots, d_n) = (1, 2, \dots, 2, 1)$

\# trees with such degrees is

$$\frac{(n-2)!}{0!\,1!\dots1!\,0!} = (n-2)!$$

Indeed, these trees are all chains from 1 to n

$$1 - \underbrace{a_1 - a_2 - \dots - a_{n-2}}_{\text{some permutation of } 2, 3, \dots, n-1} - n$$

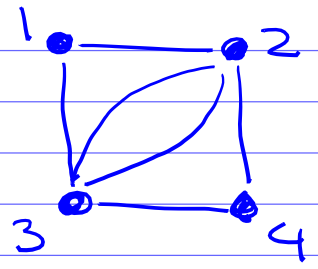
Spanning Trees

Let $G=(V,E)$ be any connected graph vertex set $V=\{1,2,\dots,n\}$ with edge set E.

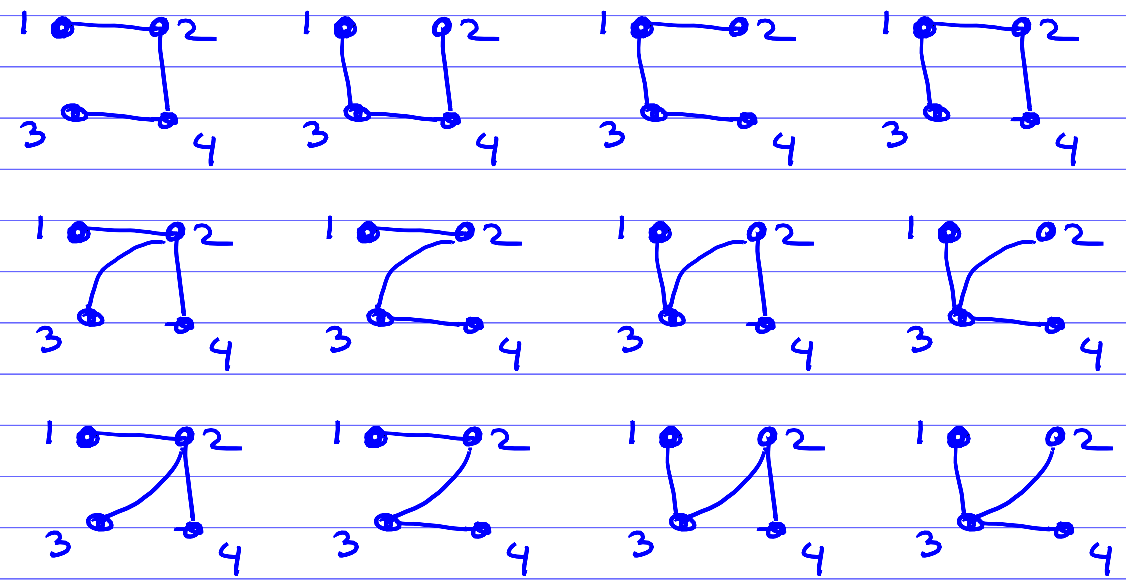
(Here we allow G to have multiple edges.)

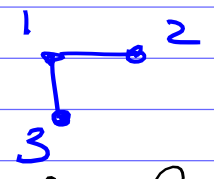
Definition. A <u>spanning tree</u> of G is a subgraph $T=(V,E')$ of G (i.e. $E'\subseteq E$) such that

- T is a tree
- T contains all vertices of G.

Example $G=$ 

Spanning trees of G:



(But 

is <u>not</u> a spanning tree of G because it does not contain all vertices of G.)

"Labelled trees on n vertices" are exactly the same thing as "spanning trees of the complete graph K_n".

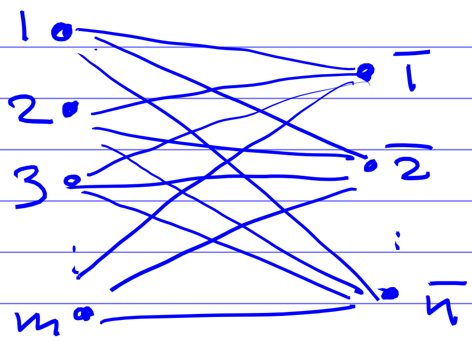
<u>Cayley's Formula</u>:

K_n has n^{n-2} spanning trees.

How about other graphs?

The <u>complete</u> <u>bipartite</u> graph:

$K_{m,n}$ =



has $m \cdot n$ edges $(i, \bar{j})$ for any $i \in [m]$, $j \in [n]$.

<u>Theorem</u> $K_{m,n}$ has exactly $m^{n-1} \cdot n^{m-1}$ spanning trees.

We can modify Prüfer coding to prove the previous formula.

Bipartite Prüfer code

T spanning tree of $K_{m,n}$ $\longmapsto (A, B)$

where $A = (a_1, \dots, a_{n-1})$, $B = (b_1, \dots, b_{m-1})$

$a_1, \dots, a_{n-1} \in \{1, 2, \dots, m\}$

$b_1, \dots, b_{m-1} \in \{\bar{1}, \bar{2}, \dots, \bar{n}\}$

- Order all vertices of $K_{m,n}$ as $1 < 2 < \dots < m < \bar{1} < \bar{2} < \dots < \bar{n}$.

- Then do Prüfer coding of T as usual, but when we remove a leaf from the first part $\{1, \dots, m\}$ of $K_{m,n}$ record the vertex to which it was attached to the sequence B; & when we remove a leaf from the second part $\{\bar{1}, \bar{2}, \dots, \bar{n}\}$ record the attached vertex to the sequence A.

<u>Exercise.</u> Prove that this construction give a bijection.

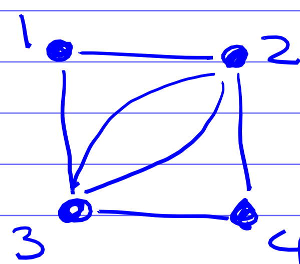
How about # spanning trees in an arbitrary graph G?

Matrix Tree Theorem

G a graph on vertices 1, ..., n.

Example

$n = 4$

$G =$ 

There are several matrices associated with G:

- Adjacency Matrix
- Incidence Matrix
- Laplacian Matrix

these are different matrices. You should not confuse them.

Adjacency Matrix $A = (a_{ij})$

$n \times n$ matrix such that

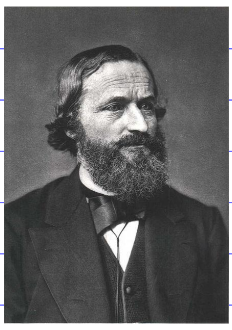
a_{ij} = # edges between vertices i & j.

Example. For the above graph we have

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 2 & 1 \\ 1 & 2 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Laplacian Matrix
(a.k.a. Kirchhoff matrix)

Pierre-Simon Laplace
(1749–1827)



Gustav Kirchhoff
(1824–1887)

$L = (l_{ij})$ $\quad n \times n$ matrix

$$L = \operatorname{diag}(d_1, \ldots, d_n) - A,$$

where $d_i = \deg_G(i)$.

Equivalently,

$$l_{ij} = \begin{cases} -\#\text{ edges between } i \text{ \& } j & \text{if } i \neq j \\ \deg_G(i) & \text{if } i = j \end{cases}$$

Example. For the above graph, we have

$$L = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 4 & -2 & -1 \\ -1 & -2 & 4 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix}$$

Some observations:

- A & L are symmetric matrices.
- All row & column sums in L are zero
- Thus $\det(L) = 0$.

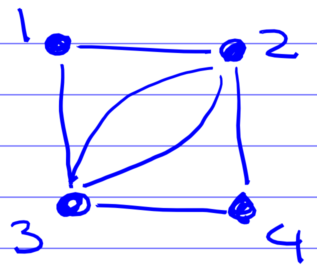
Matrix Tree Theorem

(a.k.a. Kirchhoff's theorem)

Fix $i \in \{1, \dots, n\}$.
Let $\tilde{L}$ be the $(n-1)\times(n-1)$ matrix obtained from L by removing the i^{th} row and i^{th} column.
($\tilde{L}$ is called a reduced Laplacian matrix.)

Then # spanning trees in G equals

$$\det(\tilde{L}).$$

Example $G =$ 

$$L = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 4 & -2 & -1 \\ -1 & -2 & 4 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix}$$

Take $i = 4$, $\tilde{L} = \begin{bmatrix} 2 & -1 & -1 & \cancel{0} \\ -1 & 4 & -2 & \cancel{-1} \\ -1 & -2 & 4 & \cancel{-1} \\ \cancel{0} & \cancel{-1} & \cancel{-1} & \cancel{2} \end{bmatrix}$

We get

$$\det(\tilde{L}) = \begin{vmatrix} 2 & -1 & -1 \\ -1 & 4 & -2 \\ -1 & -2 & 4 \end{vmatrix} = 12$$

For $i = 3$, we should get the same answer

$$\det \begin{bmatrix} 2 & -1 & \cancel{-1} & 0 \\ -1 & 4 & \cancel{-2} & -1 \\ \cancel{-1} & \cancel{-2} & \cancel{4} & \cancel{-1} \\ 0 & -1 & \cancel{-1} & 2 \end{bmatrix} = \begin{vmatrix} 2 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 2 \end{vmatrix}$$

$$= 2\cdot 4 \cdot 2 - 2 - 2 = 12.$$

We've seen that # spanning trees in this graph is 12.

Example $\quad K = K_n$

$$L = \begin{bmatrix} n-1 & -1 & \cdots & -1 \\ -1 & n-1 & & \\ \vdots & & \ddots & -1 \\ -1 & \cdots & -1 & n-1 \end{bmatrix} \quad \begin{matrix} n \times n \\ \text{matrix} \end{matrix}$$

$$\tilde{L} = \begin{bmatrix} n-1 & -1 & \cdots & -1 \\ -1 & n-1 & \cdots & -1 \\ \vdots & & & \\ -1 & -1 & \cdots & n-1 \end{bmatrix} \quad \begin{matrix} (n-1) \times (n-1) \\ \text{matrix.} \end{matrix}$$

Let's find all eigenvalues of $\tilde{L}$.

$$\tilde{L} - nI = \begin{pmatrix} -1 & \cdots & -1 \\ -1 & \cdots & -1 \\ -1 & \cdots & -1 \end{pmatrix}$$

has rank 1.

So its eigenvalues are all zeros, except one non-zero.

Also $\sum$ eigenvalues $=$ trace

eigenvalues of $\tilde{L} - nI$ $\qquad \underbrace{0, 0, \ldots, 0}_{n-2}, -n+1$

eigenvalues of $\tilde{L}$ $\qquad \underbrace{n, n, \ldots, n}_{n-2}, 1$

So $\det(\tilde{L}) = \underbrace{n \cdot n \cdots n}_{n-2} \cdot 1$

$$= n^{n-2}.$$