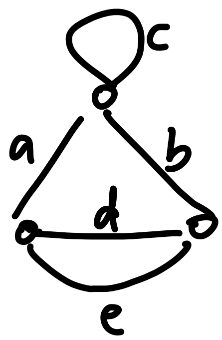
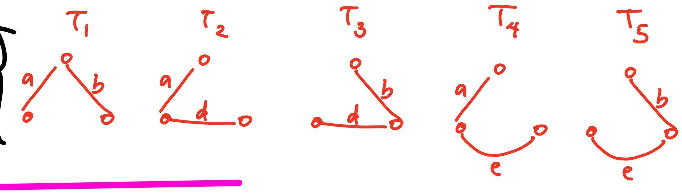
# Counting spanning trees (§2.4) and directed Euler tours (not in book)

Recall that we already defined for a multigraph $G=(V,E)$, a spanning tree for $G$ is a subset $T \subset E$ for which $(V,T)$ is a tree.

EXAMPLE: We saw $G=$




has 5 spanning trees: $\{$



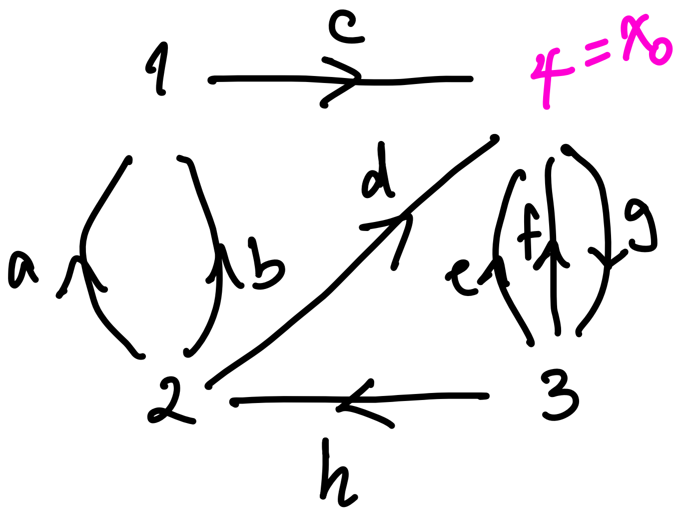

$\}$.

We'll learn how to count these, and some slightly fancier things.

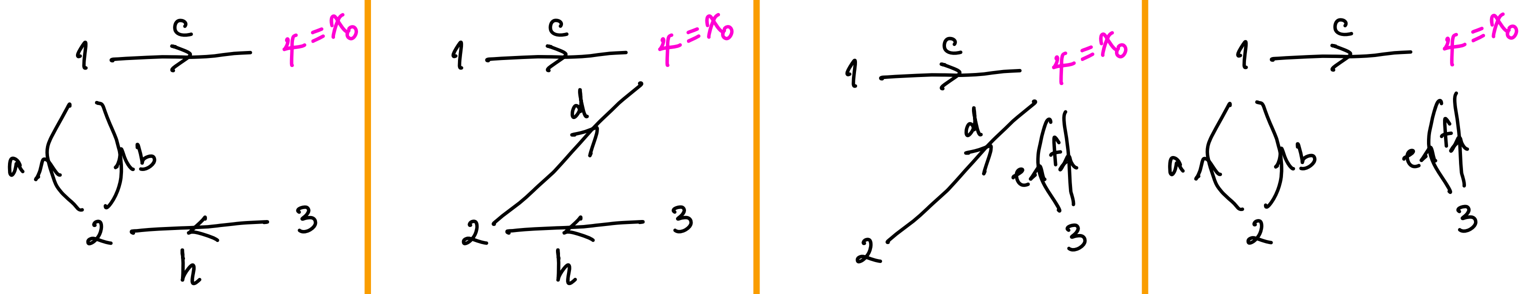
DEFINITION:
In a digraph $D=(V,A)$ with $x_0 \in V$, a spanning tree directed toward $x_0$ is a subset $T \subset A$ whose underlying undirected graph $G=(V,T)$ is a tree, and every $y \in V$ has a unique directed path $y \to \to \to \cdots \to x_0$ to $x_0$ in $D$.

## EXAMPLE

$D =$




has 9 spanning trees directed toward $x_0 = 4$:

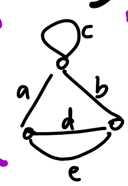



{ach, bch, cdh, cde, cdf, ace, acf, bce, bcf}

We'll compute these tree enumerators.

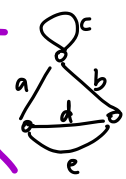
**DEFINITION:** For $G = (V, E)$ an undirected multigraph,

$$t(G) := \#\{\text{spanning trees } T \text{ for } G\}$$

e.g. $t\Big($   $\Big) = 5$

$$t(G; \underline{e}) := \sum_{\substack{\text{spanning}\\ \text{trees } T \text{ for } G}} \prod_{e \in T} e$$

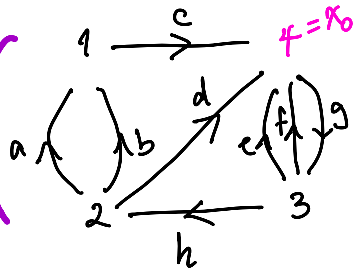
a list of variables, one for each edge in $E$

e.g. $t\Big($   $; a, b, c, d, e\Big)$

$= ab + ad + bd + ae + be$

For $D=(V,A)$ a digraph, and $x_0 \in V$,

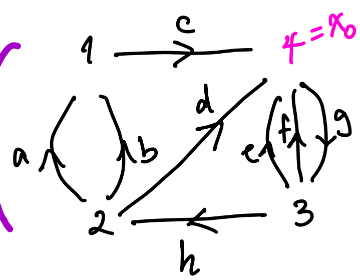
$t(D,x_0) := \#\{\text{spanning trees } T \text{ in } D \text{ directed toward } x_0\}$

e.g. $t\Big($ 

$, x_0 \Big) = 9$

$$t(D, x_0; \underline{a}) := \sum_{\substack{\text{spanning trees}\\ T \text{ in } D \text{ directed}\\ \text{toward } x_0}} \prod_{a \in T} a$$

a list of variables, one for each arc $a$ in $A$

e.g $t\Big($ 

$, x_0 \Big) =$

$ach + bch + cdh + cde + cdf + ace + acf + bce + bcf$

$= c(a+b+d)(d+e+f)$

---

Note that $t(G) = \big[t(G;\underline{e})\big]_{e=1\ \forall e \in E}$

$t(D,x_0) = \big[t(D,x_0;\underline{a})\big]_{a=1\ \forall a \in A}$,

so computing $t(G;\underline{e})$ carries more info than $t(G)$,

$t(D,x_0;\underline{a})$ carries more info than $t(D,x_0)$.

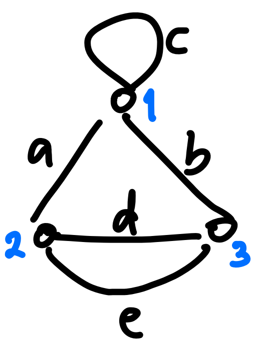
But for any $G=(V,E)$, one can also compute $t(G;\underline{e})$ from $t(\vec{G},x_0;\underline{a})$ by building $\vec{G}=(V,A)$ with one pair of antiparallel arcs $\epsilon,\epsilon'$ for each (non-loop) undirected edge $e\in E$, and then

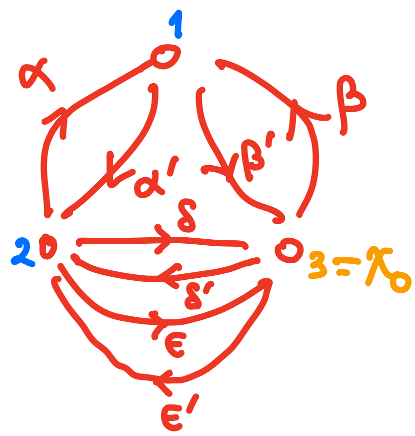
$$t(G,\underline{e}) = \Big[t(\vec{G},x_0;\underline{a})\Big]_{\substack{\epsilon=e\\ \epsilon'=e}}$$

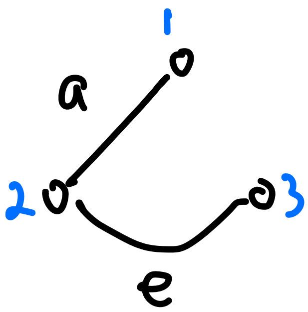
(regardless of the choice of $x_0\in V$).

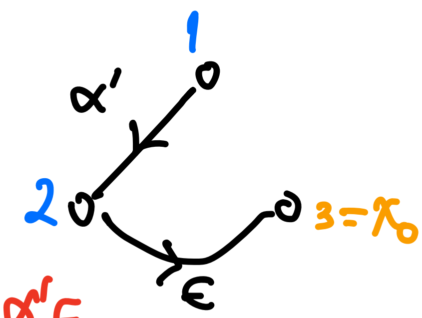
---

EXAMPLE:

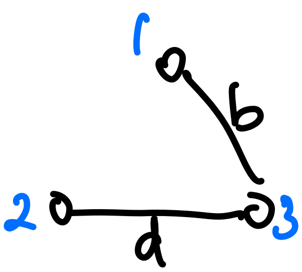
$G=$ 

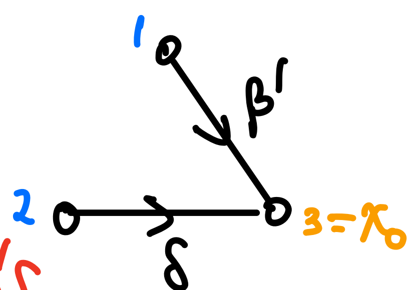

$\rightsquigarrow$ $\vec{G}=$ 


$T_1=$ 


$\longleftrightarrow$ 


$ae \longleftarrow\!\!\shortmid\ \alpha'\epsilon$

$T_2=$ 


$\longleftrightarrow$ 


$bd \longleftarrow\!\!\shortmid\ \beta'\delta$

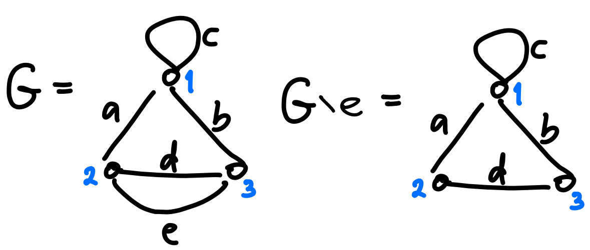
One compute $t(G)$, $t(G;\underline{q})$, $t(D,x_0;\underline{q})$ via certain recursions (although not very computationally efficient, taking $2^N$ steps if $N=|E|,|A|$), using two fundamental operations.

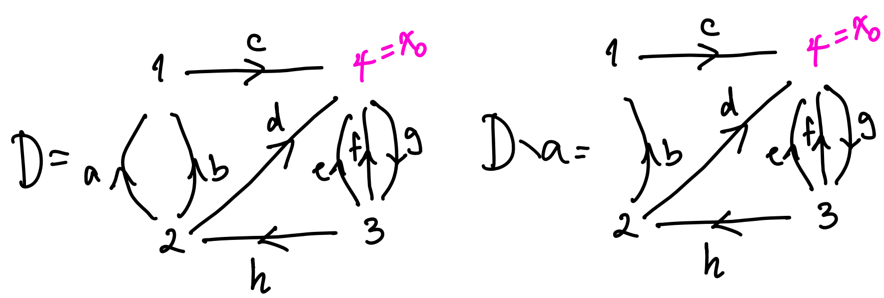
---

DEFINITION: Given $G=(V,E)$ (multigraph) or $D=(V,A)$ (digraph) and $e\in E$ or $a\in A$,

- the deletion $G\setminus e := (V\ E-\{e\})$
  $D\setminus a := (V, A-\{a\})$

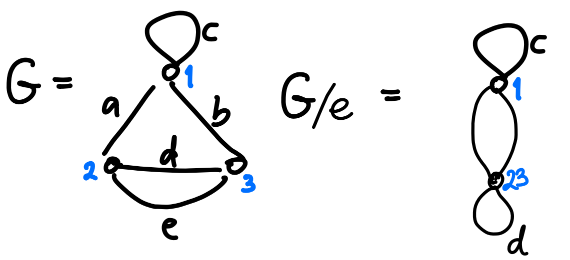
e.g.

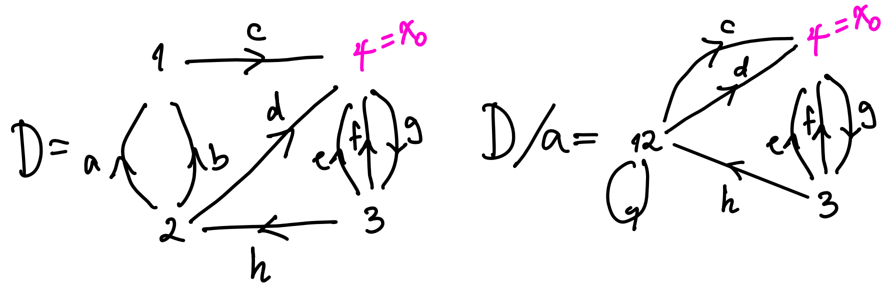



and non-loop $\underset{\{x,y\}}{\overset{\shortparallel}{e}}\in E$ or $\underset{(x,y)}{\overset{\shortparallel}{a}}\in A$, so $x\neq y$,

- the contraction $G/e := (V/_{x=y}, E-\{e\})$
  $D/a := (V/_{x=y}, A-\{a\})$



**PROPOSITION:** One can compute $t(G), t(G;\underline{e}), t(D,x_0;\underline{a})$ via these recursions and initial conditions:

(a) $1 = t\Big(\underset{G}{\boxed{\circ}}\Big) = t\Big(\underset{G}{\boxed{\circ}};\underline{e}\Big) = t\Big(\underset{D}{\boxed{\circ}}, x_0; \underline{a}\Big)$

(b) If $\hat{G} = G$ with all loops removed,
$\hat{D} = D$ with all loops removed,

then $t(G) = t(\hat{G})$, $t(G;\underline{e}) = t(\hat{G};\underline{e})$
$t(D;\underline{a}) = t(\hat{D};\underline{a})$

(c) $0 = t(G) = t(G;\underline{e})$ whenever $G$ is disconnected,
$0 = t(D, x_0;\underline{a})$ whenever $\exists\, y \in V$ with no directed path $y \to \to \cdots \to x_0$ in $D$.

(d) If $e \in E$ is non-loop, then

DELETION-CONTRACTION RECURSIONS

$$\begin{cases} t(G) = t(G\setminus e) + t(G/e) \\ t(G;\underline{e}) = t(G\setminus e, \underline{e}) + e \cdot t(G/e) \end{cases}$$

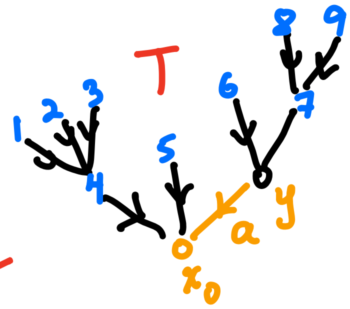
If $a \in A$ is nonloop and points toward $x_0$ ($y \xrightarrow{a} x_0$), then

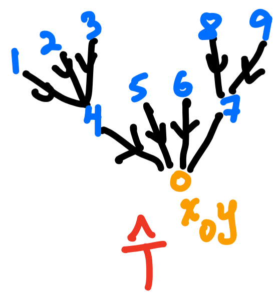
$$\left\{ t(D, x_0; \underline{a}) = t(D\setminus a, x_0; \underline{a}) + a \cdot t(D/a, x_0; \underline{a}) \right.$$

**proof:** (a), (b), (c) are all pretty straightforward.

For (d), one decomposes the set of trees $T$ as follows:

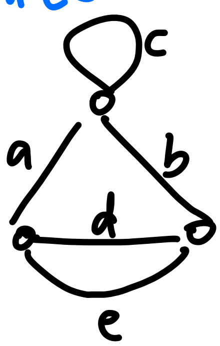
$$\left\{\begin{matrix}\text{trees } T\\ \text{in } G\\ \text{or } (D,x_0)\end{matrix}\right\} = \underbrace{\left\{\begin{matrix}\text{trees } T\\ \text{avoiding}\\ e \text{ or } a\end{matrix}\right\}}_{\begin{matrix}\| \\ \left\{\begin{matrix}\text{trees } T\\ \text{in } G\setminus e\\ \text{or } (D\setminus a,x_0)\end{matrix}\right\}\end{matrix}} \sqcup \underbrace{\left\{\begin{matrix}\text{trees } T\\ \text{using}\\ e \text{ or } a\end{matrix}\right\}}_{\begin{matrix}\updownarrow \text{ easy bijection} \\ \left\{\begin{matrix}\text{trees } \hat{T}\\ \text{in } G/e\\ \text{or } (D/a,x_0y)\end{matrix}\right\}\end{matrix}}$$



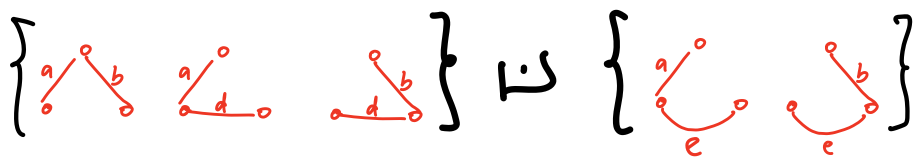




∎

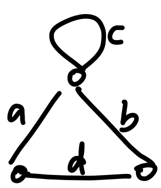
## EXAMPLE

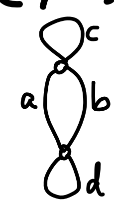
$G =$ 

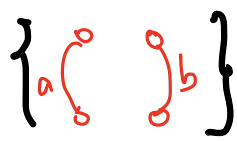

$$t(G;\underline{e}) = (ab+ad+bd) + \underbrace{(ae+be)}_{e(a+b)}$$



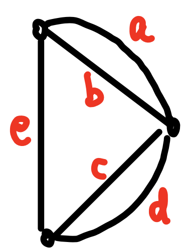

$$= t(G\setminus e;\underline{e}) + e\cdot t(G/e;\underline{e})$$

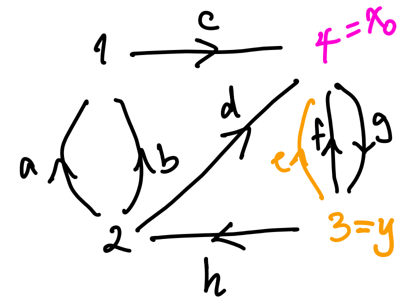



## ACTIVE LEARNING

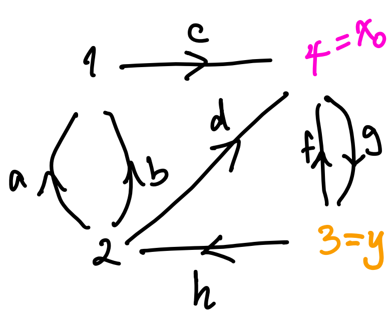
Compute $t\Bigg($  $;\underline{e}\Bigg)$ via deletion-contraction.

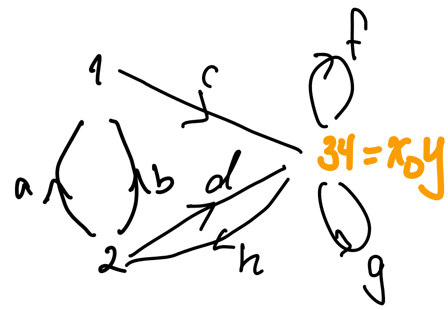
$D =$ 


$\{ach, bch, cdh, cdf, acf, bcf\} \sqcup \{cde, ace, bce\}$

$$t(D, x_0; \underline{a}) = (ach + bch + cdh + cdf + acf + bcf) + \underbrace{(cde + ace + bce)}_{e(cd + ac + bc)}$$

$$= t(D \setminus e, x_0; \underline{a}) + e \cdot t(D/e, x_0 y; \underline{a})$$



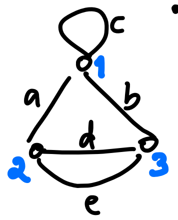




As an algorithm for computing $t(G)$, $t(G; \underline{e})$, $t(D, x_0; \underline{a})$, deletion-contraction takes $2^{|E|}$ or $2^{|A|}$ steps, so it bogs down quickly. But it helps us prove a faster method using Laplacian matrices.

**DEFINITION:** Given $G = (V, E)$ or $D = (V, A)$ the $|V| \times |V|$ Laplacian matrices $L(G)$, $L(D)$ are defined by

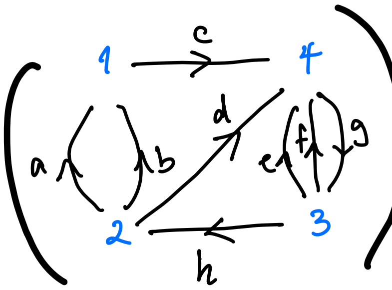
$$L(G)_{x,y} \text{ or } L(D)_{x,y} = \begin{cases} \displaystyle\sum_{\substack{\text{non-loops arcs/edges} \\ a \text{ out of } x}} a & \text{if } x = y \\ \displaystyle -\sum_{\text{arcs/edges } x \xrightarrow{a} y} a & \text{if } x \neq y \end{cases}$$

# EXAMPLES:



G

$$L(G) = \begin{array}{cc} & \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} a+b & -a & -b \\ -a & a+d+e & -(d+e) \\ -b & -(d+e) & b+d+e \end{bmatrix} \end{array}$$



D

$$L(D) = \begin{array}{cc} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} c & 0 & 0 & -c \\ -(a+b) & a+b+d & 0 & -d \\ 0 & -h & e+f+h & -(e+f) \\ 0 & 0 & -g & g \end{bmatrix} \end{array}$$

---

# THEOREM (Kirchhoff's Matrix-Tree Theorem)

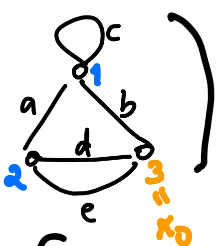
For any multigraph $G=(V,E)$ or digraph $D=(V,A)$ and any vertex $x_0 \in V$,

$$t(G;\underline{e}) = \det L(G)^{x_0,x_0} \qquad \text{where } L^{x_0,x_0} := L - \begin{Bmatrix} \text{row } x_0, \\ \text{column } x_0 \end{Bmatrix}$$

$$t(D,x_0;\underline{a}) = \det L(D)^{x_0,x_0}$$

---

# EXAMPLE:



G

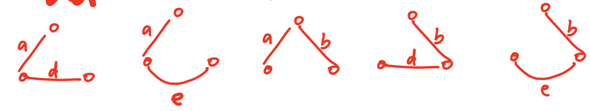
$$L(G) = \begin{array}{cc} & \begin{matrix} 1 & 2 & 3=x_0 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ x_0=3 \end{matrix} & \begin{bmatrix} a+b & -a & -b \\ -a & a+d+e & -(d+e) \\ -b & -(d+e) & b+d+e \end{bmatrix} \end{array}$$

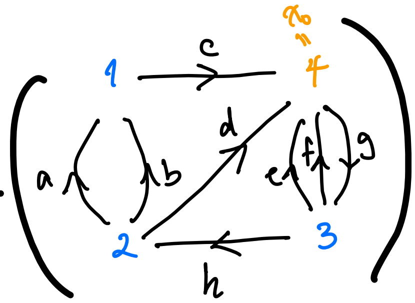
$$\det L(G)^{x_0,x_0} = \det \begin{bmatrix} a+b & -a \\ -a & a+d+e \end{bmatrix}$$

$$= (a+b)(a+d+e) - a^2$$

$$= \cancel{a^2} + ad + ae + ab + bd + be - \cancel{a^2}$$

$$= ad + ae + ab + bd + be$$



$L\Big($  $\Big) =$


D

$$\begin{array}{cc} & \begin{array}{cccc}1 & 2 & 3 & y=x_0\end{array} \\ \begin{array}{c}1\\2\\3\\x_0=4\end{array} & \left[\begin{array}{ccc|c} c & 0 & 0 & -c \\ -(a+b) & a+b+d & 0 & -d \\ 0 & -h & e+f+h & -(e+f) \\ \hline 0 & 0 & -g & g \end{array}\right]\end{array}$$

$$\det L(D)^{x_0,x_0} = \det \begin{bmatrix} c & 0 & 0 \\ -(a+b) & a+b+d & 0 \\ 0 & -h & e+f+h \end{bmatrix} = c(a+b+d)(e+f+h)$$

---

## ACTIVE LEARNING:

Explain why $\det L(D) = 0 = \det L(G)$ always.

---

## Proof of Kirchhoff's Matrix-Tree Theorem:

We'll do the proof for $D=(V,A)$, and then it follows for $G=(V,E)$ via the construction $G \rightsquigarrow \vec{G}$ from before.

The proof will show $t(D,x_0;\underline{a}) = \det\left(L(D)^{x_0,x_0}\right)$ by induction on $|A|$, by checking that $\det\left(L(D)^{x_0,x_0}\right)$ satisfies all the initial conditions (a),(b),(c) and the deletion-contraction recursion (d) that computes $t(D,x_0;\underline{a})$.

For (a), $1 = t\left(\boxed{\circ}, x_0 ; \underline{a}\right) = \det\left( [\ ] \right)$ is correct.

For (b), $t(\hat{D},x_0;\underline{a}) = t(D,x_0;\underline{a})$ if $\hat{D} = D$ with loops removed, this is consistent since $L(D) = L(\hat{D})$ (it ignores loops).

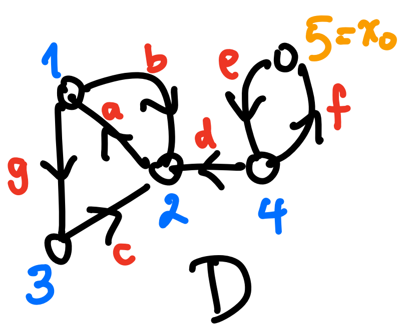
For (c), if $\exists y \in V$ with no directed path $y \to \cdots \to x_0$, consider the set $V'$ of all such $y$, and the square submatrix $L'$ of $L(D)$ indexed by rows, columns in $V'$.

Note that since $x_0 \notin V'$, this square submatrix $L'$ is contained in $L(D)^{x_0,x_0}$, and one has a block decomposition:

$$L(D) = \begin{array}{c|cc} & V' & V-V' \\ \hline V' & L' & 0 \\ V-V' & * & * \end{array}$$

Why are these entries all 0?

EXAMPLE



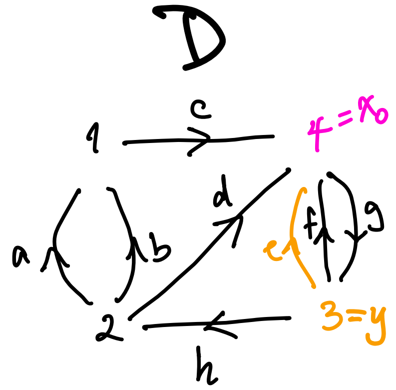

$$L(D) = \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & b+g & -b & -g & 0 & 0 \\ 2 & -a & a & 0 & 0 & 0 \\ 3 & 0 & -c & c & 0 & 0 \\ 4 & 0 & -d & 0 & d+e & -e \\ 5 & 0 & 0 & 0 & -f & f \end{array}$$

Then $\det(L') = 0$ because every row of $L'$ sums to $0$, so the all 1's vector is in its nullspace.

Therefore $\det(L(D)^{x_0,x_0}) = \underbrace{\det(L')}_{=0} \cdot \det(L(D)^{V-V'-x_0,\,V-V'-x_0}) = 0.$

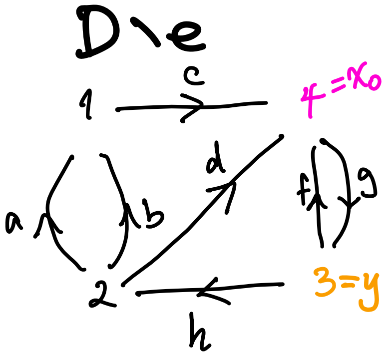
This checks (a), (b), (c) for $\det(L(D)^{x_0,x_0})$, completing the base cases for the induction on $|A|$. In the inductive step, one may assume $\exists$ some arc $e = (y, x_0)$. Consider $L(D)^{x_0,x_0}$, $L(D \setminus e)^{x_0,x_0}$, $L(D/e)^{x_0y,x_0y}$ :

**$D$**



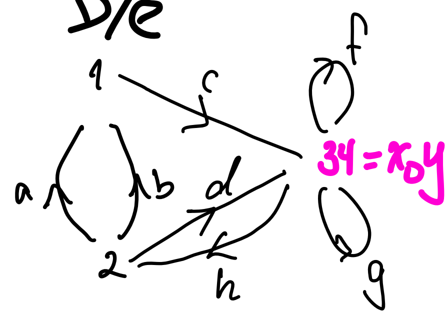

$$\begin{array}{c|cccc} & 1 & 2 & 3=y & 4=x_0 \\ \hline 1 & c & 0 & 0 & -c \\ 2 & -(a+b) & a+b+d & 0 & -d \\ y=3 & 0 & -h & e+f+h & -(e+f) \\ x_0=4 & 0 & 0 & -g & g \end{array}$$

**$D \setminus e$**




$$\begin{array}{c|cccc} & 1 & 2 & 3=y & 4=x_0 \\ \hline 1 & c & 0 & 0 & -c \\ 2 & -(a+b) & a+b+d & 0 & -d \\ y=3 & 0 & -h & f+h & -f \\ x_0=4 & 0 & 0 & -g & g \end{array}$$

**$D/e$**




$$\begin{array}{c|ccc} & 1 & 2 & 34=x_0y \\ \hline 1 & c & 0 & -c \\ 2 & -(a+b) & a+b+d & -d \\ 34=x_0y & 0 & -h & h \end{array}$$

$$\det L(D)^{x_0,x_0} = \det L(D \setminus e)^{x_0,x_0} + e \cdot \det L(D/e)^{x_0y,x_0y}$$

by expansion along the $y$ column

$$= t(D \setminus e, x_0; \underline{e}) + e \cdot t(D/e, x_0; \underline{e})$$

by induction on $|A|$

$$= t(D, x_0; \underline{e}) \quad \blacksquare$$

by (d)

# Consequences of the Matrix-Tree Theorem

① Computing the integers $t(G)$ or $t(D)$ can be done in $\leq c \cdot N^3$ steps where $N = |V|$ via linear algebra. One can use Gaussian elimination/row-reduction to compute $\det L(G)^{x_0,x_0}$ or $\det L(D)^{x_0,x_0}$ recalling that ...

- adding multiples of a row of $A$ to another leaves $\det A$ unchanged
- swapping two rows of $A$ negates $\det A$
- scaling a row of $A$ by $c$ also scales $\det A$ by $c$.

EXAMPLE:

subtract -2(row 1) from row 2 — permute 2 rows

$$\det\begin{bmatrix}2&3&4\\4&6&2\\3&4&5\end{bmatrix} = \det\begin{bmatrix}2&3&4\\0&0&-6\\3&4&5\end{bmatrix} = -\det\begin{bmatrix}2&3&4\\3&4&5\\0&0&-6\end{bmatrix}$$

$$= -2\det\begin{bmatrix}1&3/2&2\\3&4&5\\0&0&-6\end{bmatrix} = -2\det\begin{bmatrix}1&3/2&2\\0&-1/2&-1\\0&0&-6\end{bmatrix} = -2(1)(-\tfrac{1}{2})(-6)$$

$$= -6$$

scale row 1 — subtract 3(row 1) from row 2

② Occasionally one can evaluate $t(G)$ theoretically via eigenvalues of $L(G)^{x_0,x_0}$

since if $L(G)^{x_0,x_0} = P^{-1}\begin{bmatrix}\lambda_1 & & & 0\\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n\end{bmatrix}P$

then $\det L(G)^{x_0,x_0} = \det P^{-1} \cdot \det\begin{bmatrix}\lambda_1 & & 0\\ & \ddots & \\ 0 & & \lambda_n\end{bmatrix} \cdot \det P$

$= \lambda_1 \lambda_2 \cdots \lambda_n$

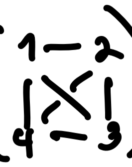
---

EXAMPLE:

THEOREM: (Cayley 1889, Borchardt 1860)

$$t(K_n) = n^{n-2}$$

e.g. $t(K_3) = t\Big($  $\Big) = 3 = 3^{3-2}$

$t(K_4) = t\Big($  $\Big) = 16 = 4^{4-2}$

proof 1: Find the eigenvalues of $L(K_n^{x_0,x_0})$

$$L(K_n) = \begin{array}{c} \\ 1 \\ 2 \\ \vdots \\ n-1 \\ n \end{array}\!\!\begin{array}{c} \begin{array}{ccccc} 1 & 2 & \cdots & n-1 & n \end{array} \\ \left[\begin{array}{cccc|c} n-1 & -1 & & -1 & -1 \\ -1 & n-1 & & -1 & -1 \\ & & \ddots & & \vdots \\ -1 & -1 & \cdots & n-1 & -1 \\ \hline -1 & -1 & \cdots & -1 & n-1 \end{array}\right] \end{array}$$

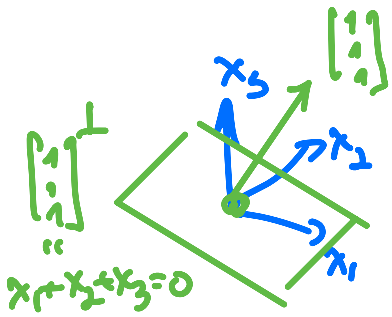
$$L(K_n^{x_0,x_0}) = \begin{array}{c} \\ 1 \\ 2 \\ \vdots \\ n-1 \end{array}\!\!\begin{array}{c} \begin{array}{cccc} 1 & 2 & \cdots & n-1 \end{array} \\ \begin{bmatrix} n-1 & & & -1 \\ & n-1 & & \\ & & \ddots & \\ -1 & & & n-1 \end{bmatrix} \end{array} = n \cdot I_{n-1} - \underbrace{\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}}_{J_{n-1}}$$

$J_{n-1} :=$ allones $(n-1)\times(n-1)$ matrix

Note $J_{n-1}$ has an eigenvector $\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$

with eigenvalue $n-1$:

$$n-1\left\{\underbrace{\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & & & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}}_{n-1}\right.\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} n-1 \\ n-1 \\ \vdots \\ n-1 \end{bmatrix} = (n-1) \cdot \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$




Also the $(n-2)$-dimensional perpendicular space

$$\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}^\perp = \left\{ \underline{x} \in \mathbb{R}^{n-1} : x_1 + x_2 + \ldots + x_{n-1} = 0 \right\}$$

lies in the nullspace (= 0-eigenspace) of $J_{n-1}$:

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & & & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ \vdots \\ x_{n-1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = 0 \cdot \begin{bmatrix} x_1 \\ \vdots \\ x_{n-1} \end{bmatrix}$$

Hence $J_{n-1}$ has eigenvalues $(\lambda_1, \lambda_2, \_\_, \lambda_{n-2}, \lambda_{n-1})$
$$= (0, 0, \ldots, 0, n-1)$$

and therefore

$L(K_n)^{x_0,x_0} = nI_{n-1} - J_{n-1}$ has eigenvalues
$$(n-\lambda_1, n-\lambda_2, \_\_, n-\lambda_{n-2}, n-\lambda_{n-1})$$
$$= (n, n, \_\_, n, 1)$$

and $\det L(K_n)^{x_0,x_0} = \underbrace{n \cdot n \cdots n}_{n-2 \text{ times}} \cdot 1 = n^{n-2}$ ■

---

proof 2: There is a beautiful bijection called Prüfer coding (1918)

$$\{\text{spanning trees } T \text{ in } K_n\} \xrightarrow{c} \{1,2,\_\,n\}^{n-2}$$
$$= \underbrace{\{(c_1, c_2, \_, c_{n-2}) : c_i \in \{1,2,\ldots,n\}\}}_{\text{a set of size } n^{n-2}}$$

$$T \longmapsto c(T) = (c_1, c_2, \ldots, c_{n-2})$$

defined by letting $c_1$ = unique neighbor of smallest labeled leaf $l_1$ of $T$

$c_2$ = neighbor of smallest leaf of $T - \{l_1\}$

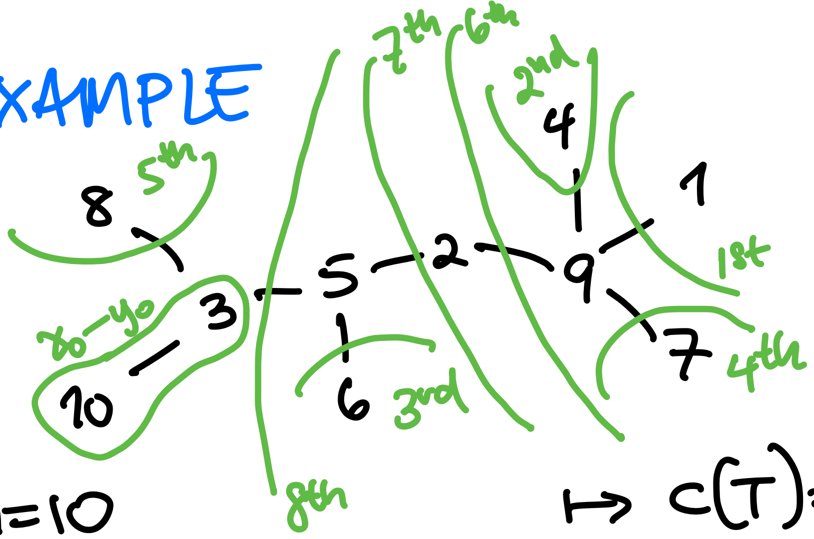
$c_3$ = neighbor of smallest leaf of $T - \{l_1, l_2\}$

$\vdots$

$c_{n-2} = \ldots.$

and stopping at the edge $T - \{l_1, l_2, \ldots, l_{n-2}\} = x_0 - y_0$

## EXAMPLE

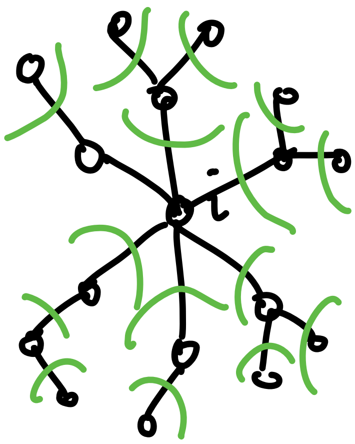
$T=$ 


$n=10$

$$\mapsto c(T) = (c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8)$$
$$= (9, 9, 5, 9, 3, 2, 5, 3)$$

---

The inverse map $c^{-1}$ takes advantage of this




**FACT:** $d_T(i) =$

$$1 + \#\{\text{occurrences of } i \text{ in } c(T)\}$$

In particular, $i$ is a leaf vertex of $T$

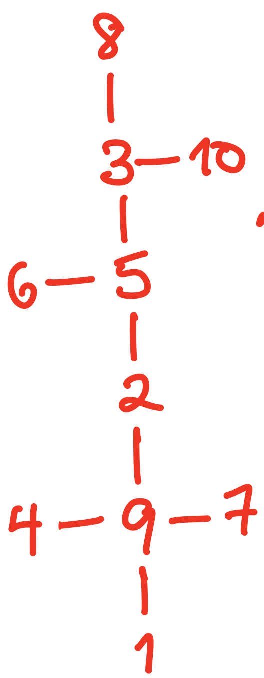
$\iff$ $i$ does not appear in $c(T)$

One uses this to recover the edges of $T$ by creating $l_i - c_i$ edges and crossing off leaves. Vertices disappearing from the code get added to the leaf list.

## EXAMPLE

| $c(T-\{l_1,l_2,\dots,l_i\})$ | leaves of $T-\{l_1,l_2,\dots,l_i\}$ |
|---|---|
| $(9,9,5,9,3,2,5,3)$ | $1,4,6,7,8,10$ |
| $(9,5,9,3,2,5,3)$ | $4,6,7,8,10$ |
| $(5,9,3,2,5,3)$ | $6,7,8,10$ |
| $(9,3,2,5,3)$ | $7,8,10$ |
| $(3,2,5,3)$ | $8,9,10$ |
| $(2,5,3)$ | $9,10$ |
| $(5,3)$ | $2,10$ |
| $(3)$ | $5,10$ |
| $()$ | $3,10$ |

$x_0 - y_0$




assemble the edges to form T

## ACTIVE LEARNING:

Someone picks a random $\underline{c}=(c_1,c_2,\dots,c_7)\in\{1,2,\dots,9\}^7$

and we'll all compute $T=c^{-1}(\underline{c})$

**proof** (that Prüfer coding is a bijection):

One can see that $c^{-1}(c(T)) = T$, but how do we know that for any $\underline{c} \in \{1,2,\dots,n\}^{n-2}$, the (multi-)graph $G$ produced by $c^{-1}$ from $\underline{c}$ really is a tree. This follows by working backwards, adding in edge $\{x_0,y_0\}$, then $\{c_{n-2}, l_{n-2}\}$, then $\{c_{n-3}, l_{n-3}\}, \dots, \{c_1, l_1\}$. At each stage, can check $l_i$ gets connected to the tree containing $\{x_0,y_0\}$ being built up, and was isolated before that. So every vertex has a path to $\{x_0,y_0\}$ and no cycles every get created. $\blacksquare$

---

**COROLLARY** (to Prüfer coding)

The number of spanning trees $T$ in $K_n$ with

$$d(T) = (d_1, d_2, \dots, d_n) \quad (\text{where } \underline{d_1+d_2+\dots+d_n = 2(n-1)})$$

Why?

is the multinomial coefficient

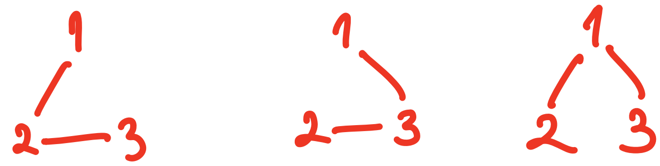
$$\binom{n-2}{d_1-1, d_2-1, \dots, d_n-1} = \frac{(n-2)!}{(d_1-1)!\,(d_2-1)!\cdots(d_n-1)!} = \begin{array}{c}\text{coefficient of} \\ x_1^{d_1-1}x_2^{d_2-1}\cdots x_n^{d_n-1} \\ \text{in } (x_1+x_2+\dots+x_n)^{n-2}\end{array}.$$

Equivalently,

$$\sum_{\text{spanning trees } T \text{ in } K_n} x_1^{d_T(1)} x_2^{d_T(2)} \cdots x_n^{d_T(n)} = x_1x_2\cdots x_n(x_1+x_2+\dots+x_n)^{n-2}.$$

# EXAMPLES

①



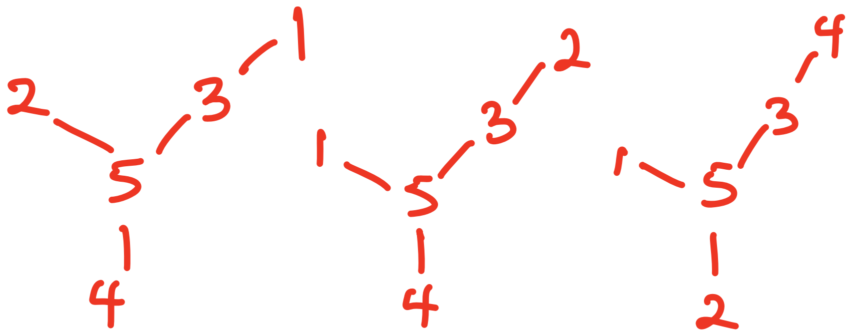
$$\sum_{\substack{\text{spanning}\\ \text{trees } T \text{ in } K_3}} x_1^{d_T(1)}\, x_2^{d_T(2)}\, x_3^{d_T(3)} = x_1x_2^2x_3 + x_1x_2x_3^2 + x_1^2x_2x_3$$

$$= x_1x_2x_3\,(x_1+x_2+x_3)^{3-2}$$

②  How many spanning trees $T$ in $K_5$ have

$$d(T) = (\overset{1}{1}, \overset{2}{1}, \overset{3}{2}, \overset{4}{1}, \overset{5}{3})\,?$$

$$\binom{3}{1-1,\ 1-1,\ 2-1,\ 1-1,\ 3-1} = \binom{3}{0,0,1,1,2} = \frac{3!}{0!\,0!\,1!\,1!\,2!} = \frac{3\cdot 2\cdot 1}{1\cdot 1\cdot 1\cdot 1\cdot 2} = 3$$



their Prüfer codes $(c_1, c_2, c_3)$ will be rearrangements of zero 1's
zero 2's
one 3
zero 4's
two 5's

i.e. $\{(3,5,5),\ (5,3,5),\ (5,5,3)\}$