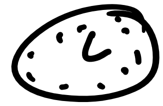
Story time 

with Slota

aka "the realities of academic writing"

Combinatorial
Scientific
Computing

(solving graph probs fast)

↳ obvious measure of quality
speed, memory, scale

Q: In practice, how much does "being the best" actually matter?

→ Very little, unfortunately

Reason: people evaluating are "human", time constraints,

"human", time constraints,
fatigue, possible mistakes

What really matters:
making everything
"idiot proof"

Not that those evaluating
are idiots → usually far
from it

Idiot proof:

- Minimizing complexity of presentation
- Stick to common formatting, terminology, basic practices
- Make it look "good", in terms of sentence structure, formatting, visually

→ nice to look at is also easier to read

Takeaway → as many famous writers and speakers have stated: "I lacked the time to write a shorter letter"

Now: Graph Theory

- TA and mentors will be grading 1000s of proofs
- think of practical consequences

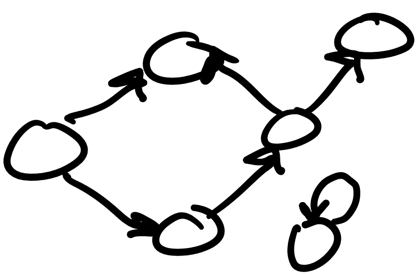
⇒ We live a practical existence constrained by fatigue, imperfections

You're goal: recognize the above. Find what you need to do and do it, not what you wish you should need to do. $\square$

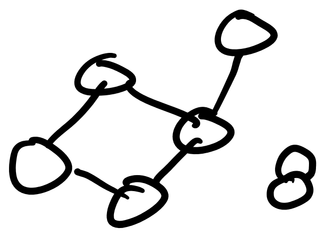
... you wish you should
need to do. $\square$

Weak and strong connectivity

Digraph D is weakly connected if the underlying graph of D is connected



D

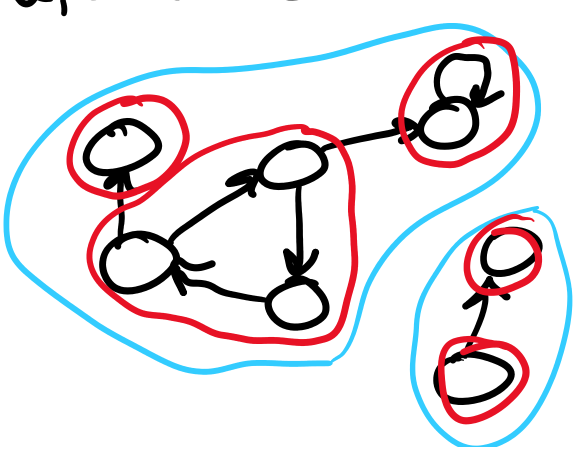


underlying graph of D

underlying graph → undirected graph that results if you remove direction from the edges

D is strongly connected if

$\forall u,v \in V(D)$: $\exists$ u,v-path



strong/weak components: maximal strongly/weakly connected subgraphs

weak components

strong components

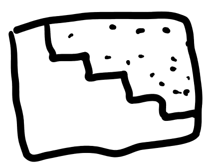
weakly connected subgraphs

Enumerating Trees

Q: How many possible graphs are there?

(simple undirected)

$$\frac{n(n-1)}{2} \leftarrow \text{possible edges}$$

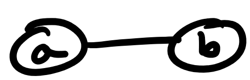


$$2^{\frac{n(n-1)}{2}} \text{ total graphs}$$

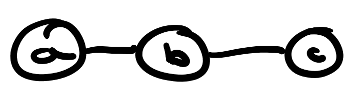
Q2: How many possible trees?

A: $n^{(n-2)}$ possible trees

(Cayley's Formula)

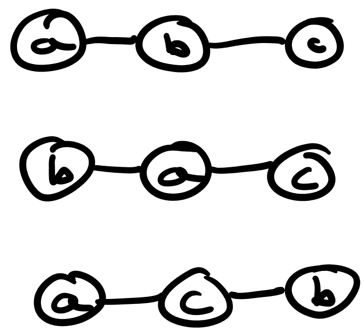
$n = 2 \rightarrow$ 

$2^{2-2} = 1$

$n = 3 \rightarrow$ 

$3^{3-2} = 3$

$n = 3 \rightarrow$



$3^{3-2} = 3$

We'll prove this formula using the notion of Prüfer codes

Prüfer code: a sequence of labels for some tree T s.t. the length of the sequence is $n-2$ and is comprised of T's vertex labels

$n = |V(T)|$

$A = \{a_1, a_2, \ldots, a_{n-2}\}$ ← prüfer code

$S = \{$vertex labels of $T\}$

sortable

$a_i \in S$

Q: How do we construct Prüfer code A for tree T?

Create Prüfer(T):

$A = \emptyset$ ← empty set

for $i = 1 \ldots (n-2)$

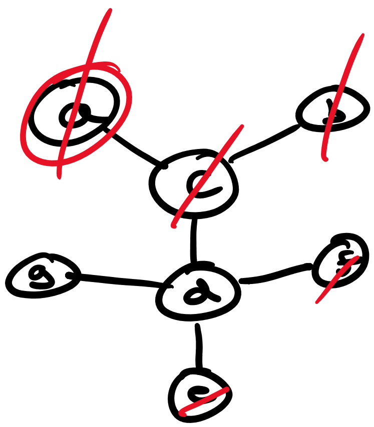
ℓ = label of the "least" remaining leaf for $i = 1 \ldots (n-2)$

ℓ = label of the "least" remaining leaf

$T = T - \ell$

a_i = remaining neighbor of ℓ

$A \leftarrow a_i$



$S = \{\cancel{a}\ \cancel{b}\ c\ d\ e\ f\ g\}$

$A = \{c\ c\ d\ d\ d\}$

Q2: Given Prüfer code A, and vertex labels S, how can we construct a tree?

CreateTree(A, S):

$V(T) = S$

$E(T) = \emptyset$

consider all $s_i \in S$ as "unmarked"

for $i = 1 \ldots (n-2)$

x = least unmarked in S that is not in $a_i \ldots a_{n-2}$

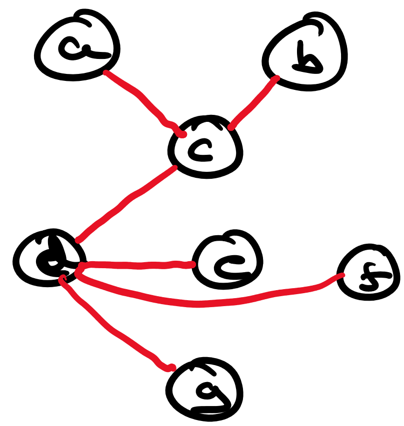
mark x in S

$E(T) \leftarrow (x, a_i)$

$E(T) \leftarrow (x, a_i)$

(x, y) = remaining unmarked in S

$E(T) \leftarrow (x, y)$



$A = \{c\ c\ d\ d\ d\}$

$S = \{a\ b\ c\ d\ e\ f\ g\}$

Observation: for a given tree T and vertex set S, we'll have a unique code A

Given code A and set S, we can construct a unique T

$f(T) = A$

bijection

Q3: How's this relate to the enumerative properties of trees/ Cayley's formula?

A3: There are $n^{(n-2)}$ ways to write a unique Prüfer code

, so there are n^{n-2} ways to
write a unique Prüfer code

$$A = \{a_1 a_2 \dots a_{n-2}\}$$

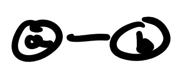
$a_i \in 1 \dots n$ possible values

Prooter code prüf
aka prooving Cayley

What we want to show: existence and uniqueness of the mapping between trees and prüfer codes

We'll do strong induction on $n = |S|$

Basis $P(2) \Rightarrow S = \{a, b\}$



$f(T) = A = \{\}$
$E(T) = (a, b)$

Consider $P(n) = T$

- Tree T with $V(T) = S$, $|S| = n$
- consider some x as the least value in S where x is a leaf in T
- consider a as neighbor of x

- consider a as neighbor of x

Construct $P(k) = T' = T - x$

$S' = S - x$

via I.H. : $A' = \{a_2 \ldots a_{n-2}\}$

$\rightarrow A'$ exists and is unique for a given (S', T')

Going from $P(k) \rightarrow P(n)$

$T' \rightarrow T$, we add back leaf x

$S' \rightarrow S$, we add back label x

$A' \rightarrow A$

=> From our Prüfer code algorithm, the vertex x and edge (x, a) would be first selected for removal, so the first a_1 value in A is guaranteed to be $\underline{a}$

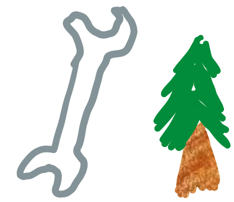
(neighbor of x)

$A = \{a, \{A'\}\}$

$f(T) = A \to$ A <u>exists</u> for our T, S and is <u>unique</u> $\square$

$\Rightarrow$ From this, it follows that Cayley's Formula holds $\square$

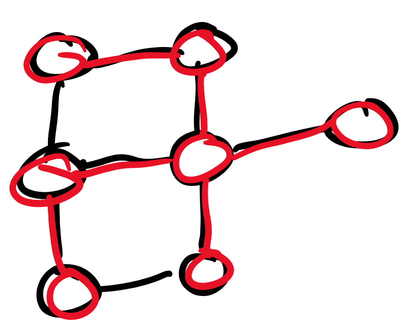
Spanning Trees



(spanner)

<u>spanning subgraphs</u> : $S \subseteq G$ is a subgraph s.t. $V(S) = V(G)$

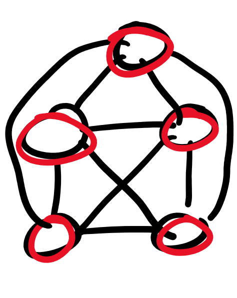
<u>spanning tree</u>: a spanning subgraph that is a tree

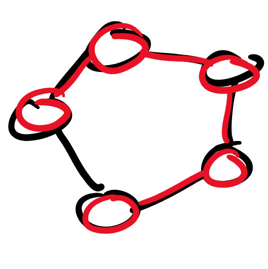


$\tau(G) = \#$ of possible spanning trees on G

$\tau(T) = 1$

$\tau(P_n) = 1$





$\tau(C_n) = n$

$\tau(K_n) = n^{(n-2)}$

Q4 (/Q7): How can we count $\tau(G)$ for some arbitrary G?

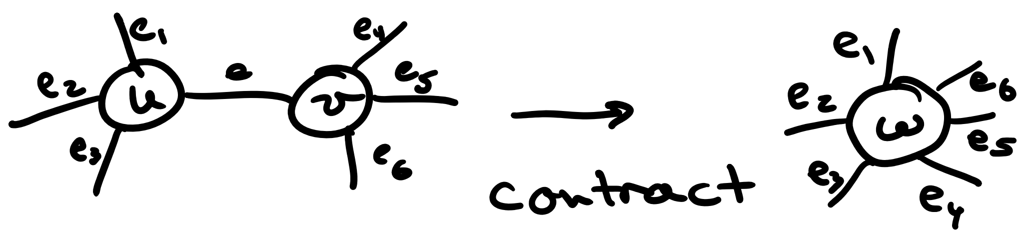
A: construct a recurrence:

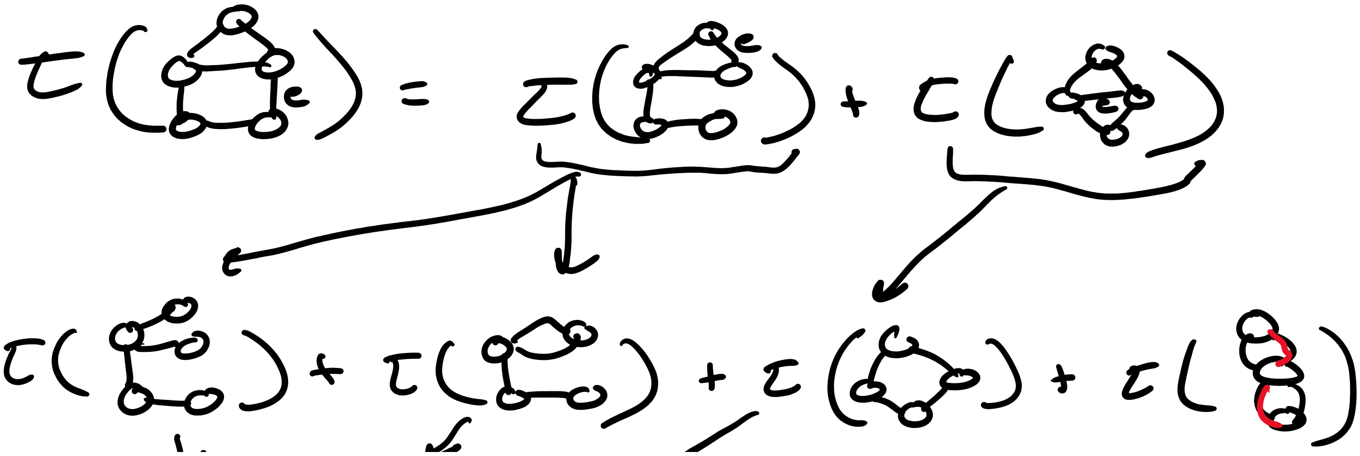
$$\underset{\#\text{ STs}}{\tau(G)} = \underset{\#\text{ ST w/o } e}{\tau(G-e)} + \underset{\#\text{ ST w/ } e}{\tau(G \cdot e)}$$

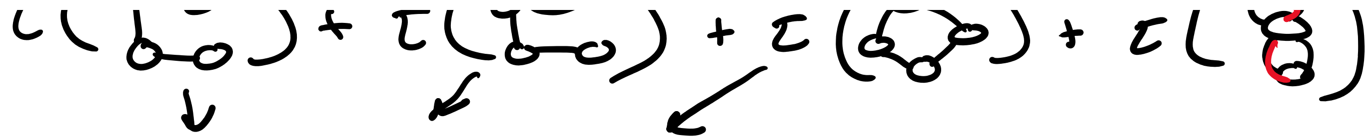
edge contraction

Edge contraction:

For edge $e = (u, v)$, we combine u, v into new vertex w that has u, v's adjacencies minus the edge e







$1 + 2 + 4 + 4 = 11$ ✓

Graceful Graphs

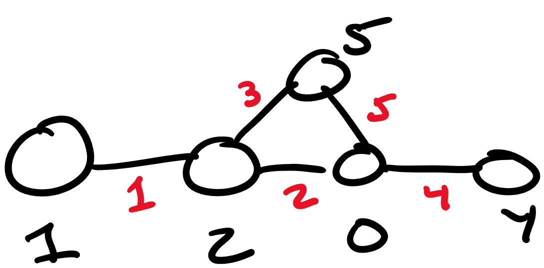
↳ graphs with a graceful labeling

Graceful labeling: labels of vertices and edge of same G s.t.

$\forall v \in V(G): L(v) = [0 \dots m = |E(G)|]$ and is unique

$\forall e \in E(G): e = (u, v)$

$L(e) = |L(u) - L(v)|$ and is unique



→ graceful ✓

Graceful Tree Conjecture:

(Ringel-Kotzig)

All trees are graceful

All Trees are graceful

(unproven)