Lecture 8 - Tree Algorithms

Thursday, February 12, 2026 9:19 AM

Joke Q: What professional football team 'toured' their way through the 1960 and 1961 AFL Championships?

Joke A: Houston Eulers

Graceful graphs

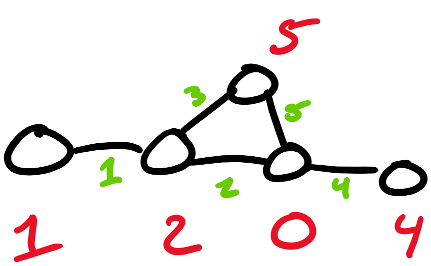
↳ graphs with a graceful labeling

<u>Graceful labeling</u>: labels of vertices and edges of some G s.t.

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

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

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



→ Graceful ✓

why do we care?

A: ????

A: ????

Graceful Tree Conjecture

(Ringel-Kotzig)

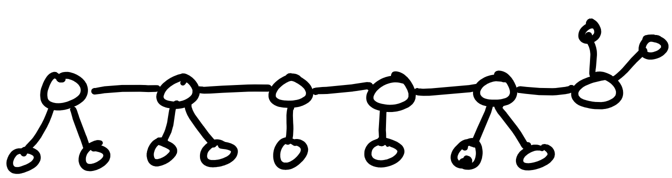
All trees are graceful

(unproven)

However: a couple of tree classes have been proven

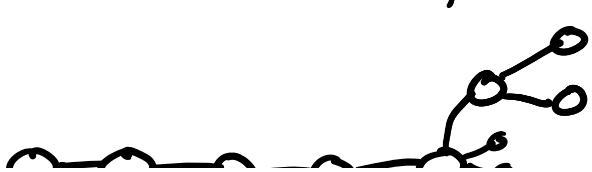
Caterpillar graphs:

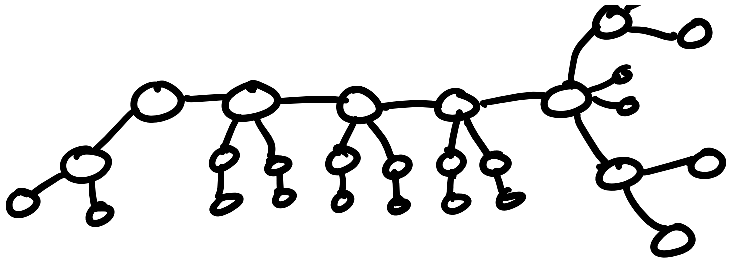
Tree graph where all vertices are at most 1-hop from a single path



Lobster Graphs

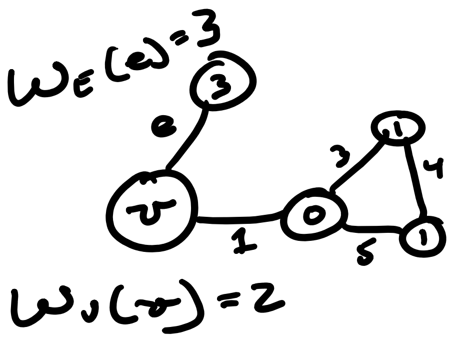
Same as above, but 2-hops





Weighted Graphs

Weighted $G = \{V, E, W_V, W_E\}$



W_V = vertex weight set

$|V| = |W_V|$

W_E = edge weight set

$|E| = |W_E|$

Note: Weighted graphs can have vertex and/or edge weights

- Can be integer/real/imaginary
- Can be positive/negative
- Can have multiple weights

Underlying Assumption

For this class: positive integer weights

For this class: positive integer weights (usually on edges)

Minimum spanning tree (MST)

↳ a spanning tree on an edge-weighted G s.t. the sum of the weights of edges in the spanning tree are minimized

G 

weight of MST = 10

To get a MST: Krushkal's Algo.

MST ($G = \{V, E, \omega\}$) ← edge weights

assume connected

$V(T) = V(G)$ ← output

$E(T) = \emptyset$

Sort ω, E in nondecreasing order

For all $\omega(e), e \in \omega, E$

For all $w(e), e \in W, E$

if numComps$(T+e) <$ numComps(T):

outputs # of components

$E(T) \leftarrow e$

if numComps$(T) == 1$

Break

Prove correctness of Krushkal

show: M, S, T

T: any edge we add decreases the number of componets

→ every edge is a cut edge

→ no edge is on a cycle

→ end at 1 component (connected)

=> we have a tree ✓

S: We continue until T is connected

→ we start with $V(T) = V(G)$

=> T spans G ✓

M: pseudo-algorithmic argument

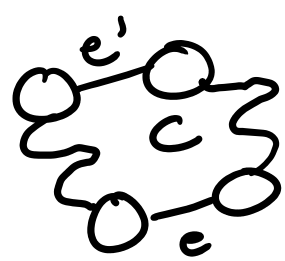
Consider: Krushkal outputs a ST

Consider: Krushkal outputs a ST
that is <u>not</u> optmal

define: T^* = actual MST (hypothetical)
T = output of Krushkal

consider some $e \in E(T)$ s.t. $e \notin E(T^*)$
where e is the first such edge
output by the algorithm

Note: adding e to T^* creates a cycle
→ consider $e' \in C$, $e' \notin E(T)$



Note: T^* has all edges in T that
were selected before e

→ both e and e' were available
for selection

→ $w(e) \leq w(e')$

define: $T' = T^* + e - e'$

→ $W(T') \leq W(T^*)$
(sum of weights)

→ T' has one more edge in common with T than T^*

⇒ repeat this for all possible edges, transforming $T^* \to T$

⇒ Hence, as $W(T) \leq W(T^*)$, T is optimal □

To prove Prims: (grow tree using min edge weights)

→ same basic approach

Note: considers edges based on relative edge weights

Single source shortest paths aka SSSP

Consider some edge-weighted G

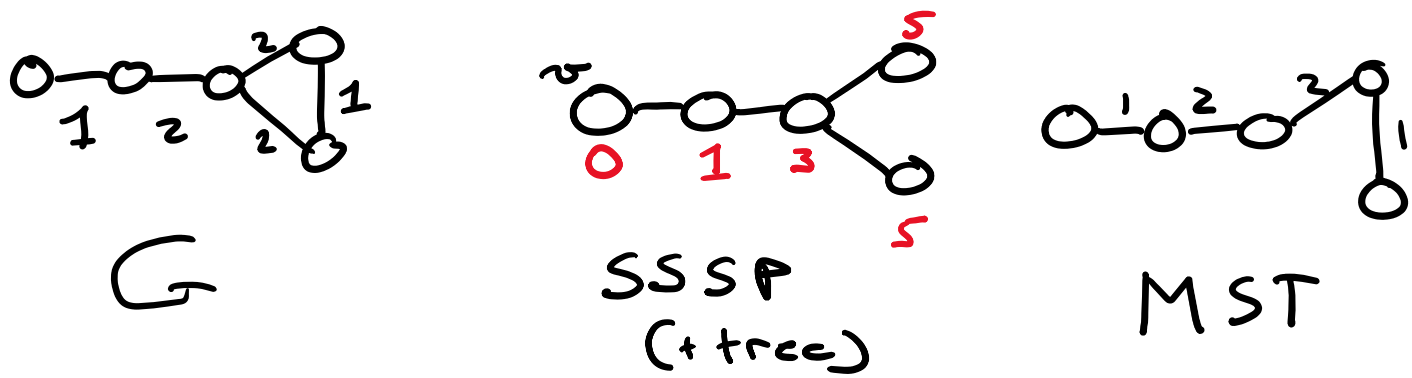
SSSP: from some 'root' $v \in V(G)$, we want to identify $d(v, u)$

we want to identify $d(v, u)$ for all $u \in V(G)$

sum of edge weights

AND

consider all u, v-paths as a shortest paths tree



Note: SSSP is not necessarily equal to a MST

Dijksta's Algorithm

SSSP$(G = \{V, E, \omega\}, u)$ root

$\forall v \in V(G) : d(u, v) = \infty$ infinity

$d(u, u) = 0$

unvisited set → $S = V(G)$

while $S \neq \phi$:

$\omega = \min d(u, v)$

while $S \neq \varnothing$

$$w = \min_{v \in S} d(u, v)$$

$\forall x \in N(w)$ s.t. $x \in S$:

$t = w((x, w))$ ← weight of edge

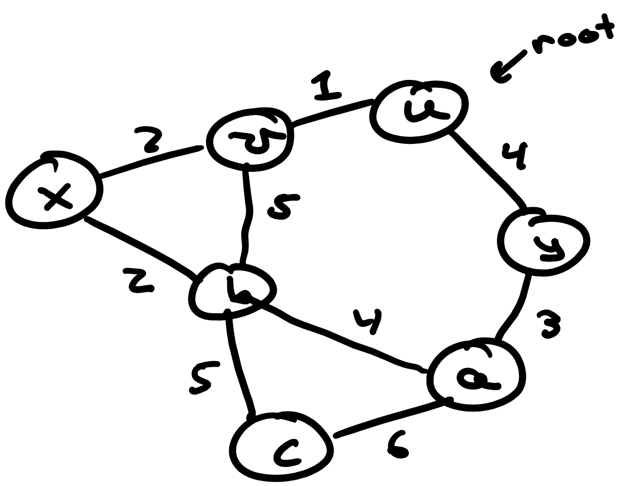
if $d(u,w) + t < d(u,x)$

$d(u,x) = d(u,w) + t$

(to output SSSP, keep track of optimal edges used)

$S = S - w$

Dijkstra in action



S	t_0	t_1	t_2	t_3
$\boxed{u}$	0	0	0	0
$\boxed{v}$	∞	1	1	1
$\boxed{x}$	∞	∞	3	3
y	∞	4	4	4
a	∞	∞	∞	∞
b	∞	∞	6	5
c	∞	∞	∞	∞

...

Dijkstra Correctness Proof

What to show:

At every iteration, show

What to show

At every iteration, show

Assume X is set of visited vertices

$\forall v \in X \quad D(v) = d(u,v)$

output from alg.

$\forall v \notin X \quad D(v)$ is shortest path to v through X

We'll do weak induction on $|X|$

Basis $P(1)$: $X = \{u\}$ and $D(u) = d(u,u) = 0$

all $v \in N(u)$ have weight of edge (u,v) as $D(v)$

$P(k) \rightarrow |X| = k$

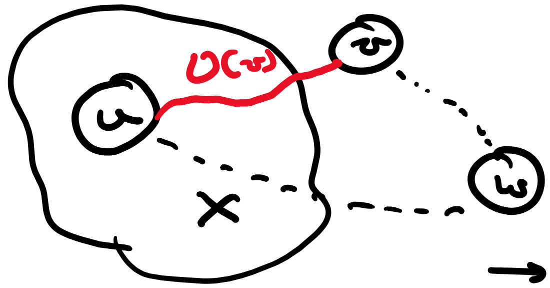
assume via I.H. that the two conditions above hold

$P(k+1) \rightarrow X' = X + v$

v is selected s.t. $D(v)$ is least for all $v \notin X$

First show: $D(v) = d(u,v)$

via I.H. $\longrightarrow$ shortest path from X to v is $D(v)$, so any possible to v is $D(v)$, so any possible path that exits X and reaches v is bounded by $D(v)$



$D(w) \geq D(v)$

$\rightarrow$ so hypothetical path can't exist ✓

Secondly, show: $D(y)$ is correct for all $y \notin X'$ where $X' = X + v$

By I.H. $\rightarrow$ $D(y)$ is shortest u,w-path distance from X

we update $D(y) = \min(\underbrace{D(y)}_{\text{distance through } X}, \underbrace{D(v) + w((v,y))}_{\text{distance through } X'})$

$\Rightarrow$ shortest possible path to y through X', as v is the only way to y through a vertex not originally in X ☑

vertex not originally in X $\square$