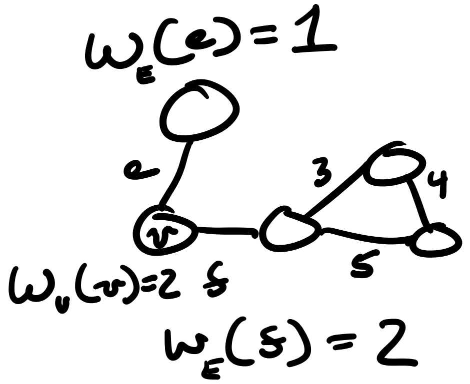
Irony: babbling incoherently for 10 minutes on the importance of clear and effective communication.

Weighted graphs

weighted $G = \{V, E, W\}$



$W = W_V$ vertex weights

$W = W_E$ edge weights

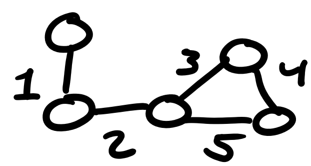
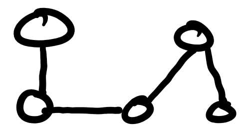
$W = \{W_V, W_E\}$ both

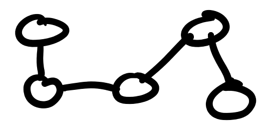
Note: weighted graphs can have vertex and/or edge weights

- Can be inter/real/imaginary
- Can be positive/negative
- Weights for vertices/edges are defined for all vertices/edges in the graph
  $|W_V| = |V|$, $|W_E| = |E|$
- Can have multiple weights per edge or vertex

Minimum spanning tree (MST)

→ a spanning tree on an edge-weighted graph that has the minimum sum of weights

MST of G  



To get an MST: Kruskal's Algorithm

output ↓ $V(T) \leftarrow V(G)$ ← input

$E(T) = \emptyset$

sort w, E in nondecreasing order

for all $w, e \in W, E$

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

  ↑ calculates number of comps.

    $E(T) \leftarrow e$

  if numComps $(T) = 1$:

    break

Prove Correctness of Kruskal's

Prove Correctness of Kruskal's

To show: prove M, S, T

T: any edge we add decrease the number of components

→ every edge is a cut edge

→ no edge is on a cycle

→ we end with a connected T

S: Assuming G is connected, we go until T is a single component

→ T contains all $v \in V(G)$

→ T is a spanning subgraph

M: pseudo-algorithmic argument

Consider: Kruskal outputs a ST (spanning tree) that is not minimum

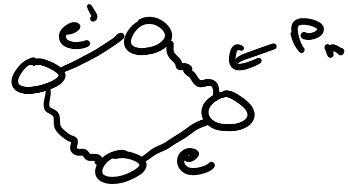
define: T^* = actual MST (hypothetical)

T = output of Kruskal

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

Adding e to T^* creates cycle C

→ 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 by T: $w(e) \leq w(e')$

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

Note: $w(T') \leq w(T^*)$

→ T' has more edges in common with T than T^*

=> repeat this process for all differing edges then $T' \to T$

differing edges then $T' \to T$
and therefore $T^* \to T$ $\square$

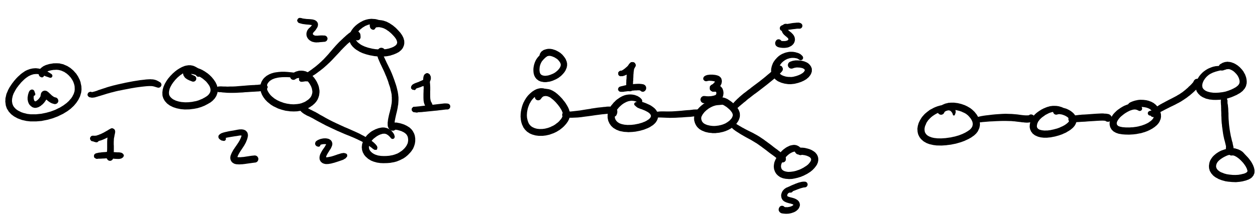
To prove Prim's (from notes)
→ do the same as above

Single source shortest paths (SSSP)

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

AND

Consider all u,v paths as a shortest paths tree



G   SSSP tree   MST

Note: SSSP tree $\neq$ MST

Dijkstra's Algorithm

Dijkstra's Algorithm

for the SSSP problem

$\forall v \in V(G): \underbrace{D_u(v)}_{} = \infty$ ← input; output distances

$D(u) = 0$ ← root

$S = V(G)$ ← unvisited set

while $S \neq \emptyset$:

$\quad w = \min_{v \in S} D(v)$ ← vertex in S with smallest current distance

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

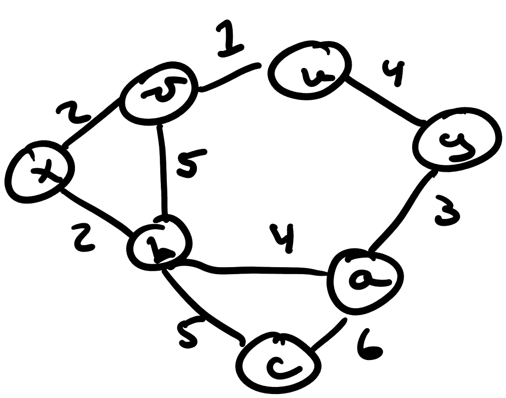
$\quad\quad t = W(x, w)$ ← weight of edge

$\quad\quad$ if $D(w) + t < D(x)$

$\quad\quad\quad D(x) = D(w) + t$

$\quad S = S - w$

Dijksta in action



	D_0	D_1	D_2	D_3	D_4
$\boxed{u}$	0	0	0	0	...
$\boxed{v}$	∞	1	1	1	
$\boxed{x}$	∞	∞	3	3	
y	∞	4	4	4	
a	∞	∞	∞	∞	
b	∞	∞	6	5	
c	∞	∞	∞	∞	

exercise 4 reader

Dijkstra Correctness Proof

what to show:

At every iteration:

X: visited set of vertices

1 - $\forall v \in X \quad D(v) = d(\overset{\text{root}}{u}, v)$

2 - $\forall v \notin X \quad D(v)$ 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 $v \in N(u)$ take the weight of edge (u,v) as $D(v)$

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

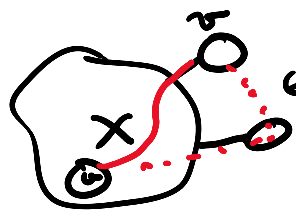
assume via I.H. that the two conditions hold

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

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

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

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



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

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

We update $D(w) = \min\left(\underbrace{D(w)}_{\text{distance from } X}, \underbrace{D(v) + W(v,w)}_{\text{distance from } X'}\right)$

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

a vertex not originally in X $\square$