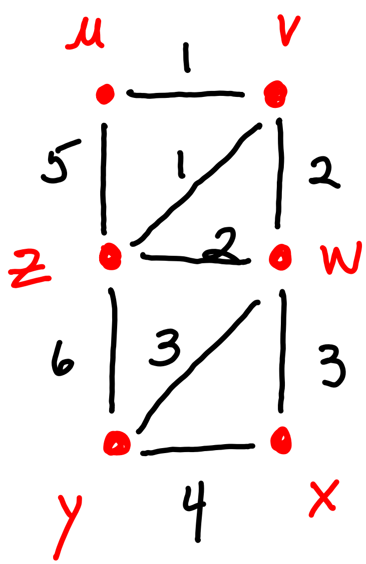
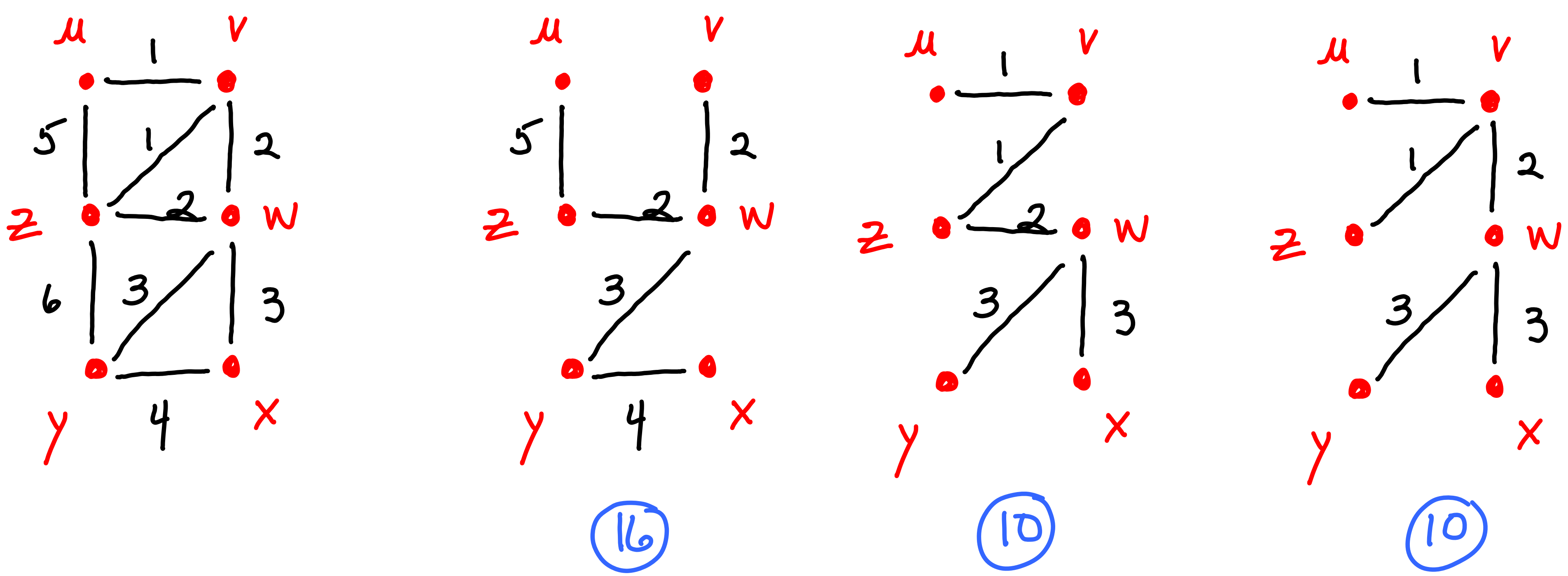
Weighted graph - weights assigned to edges

Minimum spanning tree - one that minimizes sum of edge weights



Weighted graph - weights assigned to edges

Minimum spanning tree - one that minimizes sum of edge weights



Kruskal's MST Algorithm

$G = (V, E)$ (connected) $\quad |V| = n$

$H \leftarrow (V, \emptyset)$

while $|E(H)| < n-1$ do

- $e \leftarrow$ min weight edge of G s.t. $H+e$ is acyclic
- add e to H

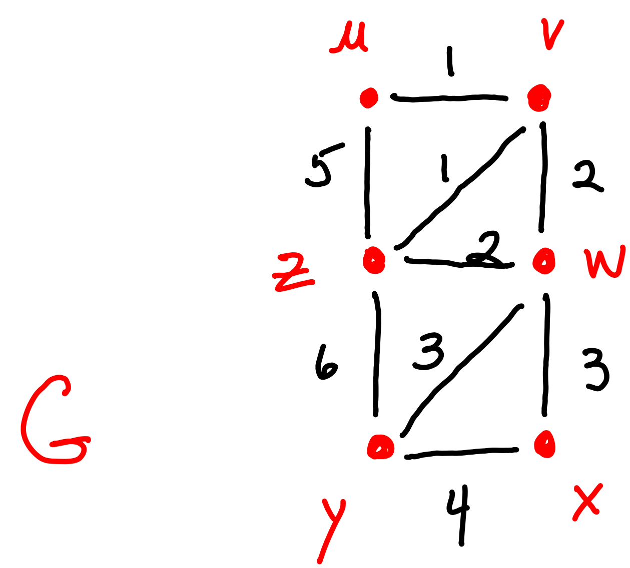
Kruskal's MST Algorithm

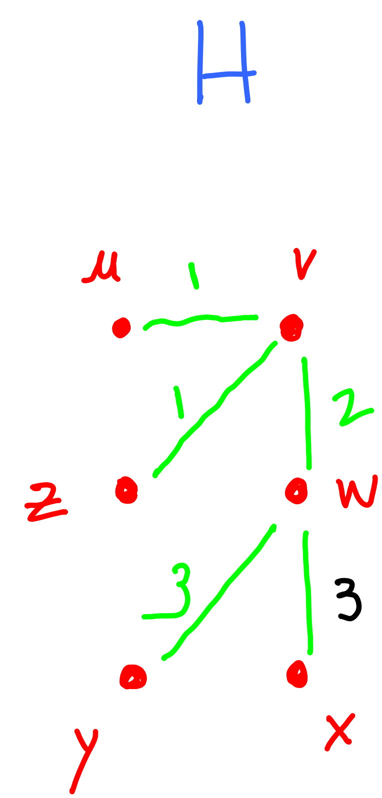
$G=(V,E)$ (connected) $\quad |V|=n$

$H \leftarrow (V, \emptyset)$

while $|E(H)| < n-1$ do

- $e \leftarrow$ min weight edge of G s.t. $H+e$ is acyclic
- add e to H





Proof of Kruskal Let T be the tree produced by Kruskal.

Label the edges of T: $e_1, e_2, e_3, \ldots$ in order added by Kruskal

Let T^* be a MST.

Proof of Kruskal

Let T be the tree produced by Kruskal.

Label the edges of T: $e_1, e_2, e_3, \ldots$ in order added by Kruskal

Let T^* be a MST. First show that if all edge weights are distinct, then $T = T^*$.

Proof of Kruskal

Let T be the tree produced by Kruskal.

Label the edges of T: $e_1, e_2, e_3, \ldots$ in order added by Kruskal

Let T^* be a MST. First show that if all edge weights are distinct, then $T = T^*$. If not, let e_j be the first edge of T not in T^* (then $j \leq n-1$)

Proof of Kruskal   Let T be the tree produced by Kruskal.

Label the edges of T: $e_1, e_2, e_3, \ldots$ in order added by Kruskal

Let T^* be a MST. First show that if all edge weights are distinct, then $T = T^*$. If not, let e_j be the first edge of T not in T^* (then $j \leq n-1$)

By Prop. 2.1.7, there is an edge e' in $E(T^*) - E(T)$ s.t. $T^* + e_j - e'$ is a spanning tree of G.

Proof of Kruskal   Let T be the tree produced by Kruskal.

Label the edges of T: $e_1, e_2, e_3, \ldots$ in order added by Kruskal

Let T^* be a MST. First show that if all edge weights are distinct, then $T = T^*$. If not, let e_j be the first edge of T not in T^* (then $j \leq n-1$)

By Prop. 2.1.7, there is an edge e' in $E(T^*) - E(T)$ s.t. $T^* + e_j - e'$ is a spanning tree of G.

But since T^* is a MST, $w(e_j) \geq w(e')$.

On the other hand, note Kruskal chose e_j over e'.

But all of the edges $e_1, e_2, \ldots, e_{j-1}, e'$ are in T^*.

So e' does not create a cycle with $e_1, e_2, \ldots, e_{j-1}$.

I.e, e' was eligible to be chosen by Kruskal at the same time as e_j. But it did not get chosen, so $w(e_j) \le w(e')$. Thus $w(e_j) = w(e')$, contradicting that all weights are distinct.

On the other hand, note Kruskal chose e_j over e'.

But all of the edges $e_1, e_2, \ldots, e_{j-1}, e'$ are in T^*.

So e' does not create a cycle with $e_1, e_2, \ldots, e_{j-1}$.

I.e, e' was eligible to be chosen by Kruskal at the same time as e_j. But it did not get chosen, so $w(e_j) \leq w(e')$. Thus $w(e_j) = w(e')$, contradicting that all weights are distinct.

If edge weights <u>not</u> distinct, note

replace T^* by $T^* + e_j - e'$ and repeat

$\blacksquare$

★★★ for advanced students only ★★★

instead of:

If edge weights not distinct, note

replace T^* by $T^* + e_j - e'$ and repeat

DO THIS:

proof by extremality: Let T^* be a MST which contains $e_1, e_2, \ldots, e_{j-1}$ for largest possible j.

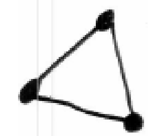
Then show that if $T^* \neq T$ then you can construct $T^* + e_j - e'$ to reach a contradiction.

. <u>**distance**</u> from u to v in G, $d_G(u,v)$:

least length of a u, v path in G, if one exists.

<u>**diameter**</u> of G:

$$\max_{u,v \in V(G)} d(u,v)$$



<u>**eccentricity**</u> of vertex u of G:

$$\epsilon(u) = \max_{v \in V(G)} d(u,v)$$

<u>**radius**</u> of G:

$$\min_{u \in V(G)} \epsilon(u)$$

Shortest Path Problems

Each edge e has a real weight $w(e)$.
Find **minimum weight path** joining pairs of vertices.

All Pairs [Floyd 1962]

Single Source [Dijkstra 1959]

Single Pair

Problems with edges of negative weight:

[Dijkstra] may fail in the presence of negative weight edges.

[Floyd] will work for negative weight edges, but not for negative weight cycles.

Dijkstra's Algorithm

G - weighted graph or digraph; (NO NEGATIVE WEIGHT:

Let $w(x, y) \quad = \quad$ weight of xy if $xy \in E(G)$;
∞, otherwise.

Source $u \in V(G)$

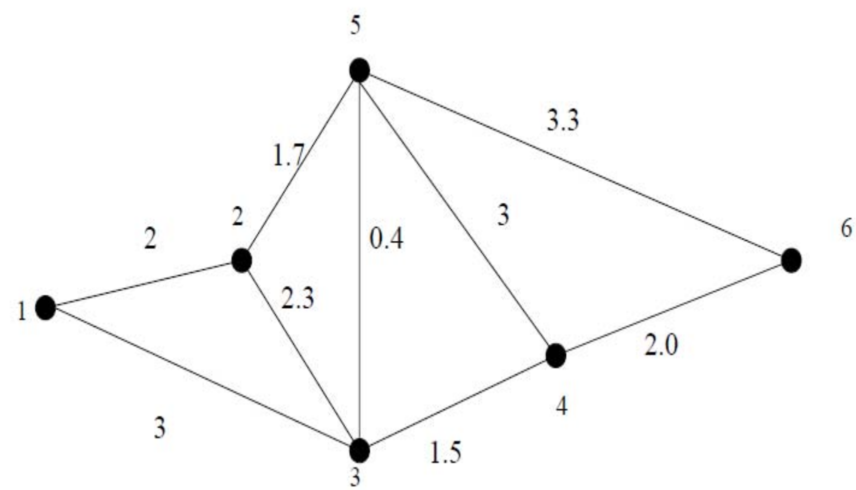
Maintain set S of vertices to which minimum-weight path from u is known.

Maintain, for each $z \in V(G) - S$, a tentative weight $t(z)$ from u initialized to $w(u, z)$.

As long as $S \neq V$ do the following:

> **Find vertex $v \notin S$ for which $t(v)$ is minimum.**
>
> **Add v to S.**
>
> **For each $z \notin S$ which is adjacent to v, update tentative weight:**
>
> $$t(z) \ \leftarrow \ \min\{t(z), t(v) + w(v, z)\}$$

Example: Find minimum weight paths from vertex 1 using Dijkstra's algorithm.



Proof of Correctness

CLAIM: At beginning of each iteration,

$t(v) =$

(a) weight of cheapest path from u to v, if $v \in S$;

(b) otherwise, weight of cheapest path from u to v using only vertices from S as intermediate vertices.

PROOF OUTLINE: (Induction on iteration)

Basis: Claim is true at beginning of iteration 1.

Ind. Assume claim is true at beginning of iteration i. Show it is still true at end. Let v be vertex chosen during iteration i.

(a) If a cheaper u, v path contained a vertex not in S, let x be the first such vertex on this path. Then $t(x) < t(v)$, a contradiction.

(b) Check that t-values for $z \notin S$ are correctly updated.