

## Minimum Spanning Tree

Weighted graphs:  $G = (V, E)$  where each  $e \in E$  has weight  $w(e) \in \mathbb{Q}$ .

Example:  $V = \{\text{servers}\}$

$E = \{\text{pairs of servers that can be connected}\}$

$w(e)$ : cost of connecting the endpoints of  $e \in E$ . (non-negative)

Problem: What is the cheapest way to connect all the servers?

In other words, find a subgraph  $H$  of  $G$  with min weight.

$$H = (V, E_H) \text{ where } E_H \subseteq E. \quad \text{and} \quad w(H) = \sum_{e \in E_H} w(e).$$

Observation: There is always a min weight  $H$  that is acyclic (i.e. a tree).

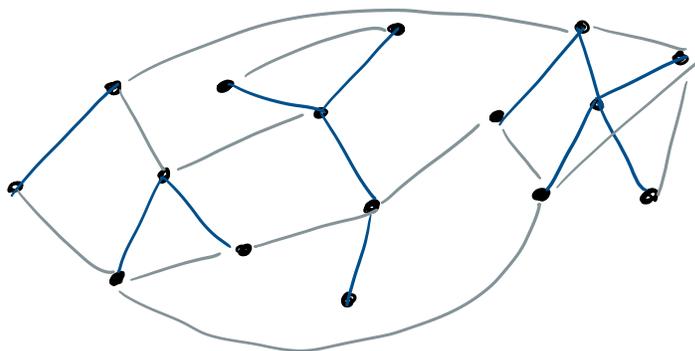
Can remove any edge from a cycle and not increase weight.

A spanning tree  $T = (V, E_T)$  of  $G$  is a connected acyclic subgraph of  $G$ .

- a)  $T$  is a tree      b)  $T$  is a subgraph of  $G$

MST problem: given weighted undirected  $G = (V, E)$ , find a spanning tree of  $G$  with minimum  $w(T)$ .

# of spanning trees of  $G$  can be as large as  $n^{n-2}$ .



Algorithm template:

Start with  $F = (V, E_F)$ ,  $E_F = \emptyset$

While  $F$  is not spanning tree:

Find an edge  $(u, v) \in E$ , such that:

- a)  $u$  and  $v$  are in different CC of  $F$
- b)  $E_F \cup \{(u, v)\}$  is contained in some MST of  $G$

Add  $(u, v)$  to  $E_F$

Thm: Let  $F = (V, E_F)$  be a spanning forest of  $G = (V, E)$  s.t.  
 $E_F \subseteq E_T$  for some MST  $T = (V, E_T)$  of  $G$ .

Let  $V_1, \dots, V_k$  be connected components of  $F$ .

For any  $i$ , if  $u \in V_i$ ,  $v \notin V_i$ ,  $(u, v) \in E$ , and

$$w(u, v) = \min \{ w(u', v') : u' \in V_i, v' \notin V_i, (u', v') \in E \}$$

then  $E_F \cup \{(u, v)\} \subseteq E_{T^*}$  for some MST  $T^*$  of  $G$ .

Fact: If  $T = (V, E_T)$  is a spanning tree of  $G$ , then  $T' = (V, E_T \cup \{(u, v)\})$ ,  
 $e = E \setminus E_T$  has an unique cycle containing  $e$ .

If  $e = (u, v)$  then  $u$  and  $v$  are connected by exactly one path in  $T$ .

Proof of Thm:

If  $(u,v) \in E_T$ , then we are done (take  $T^* = T$ ).

Assume  $(u,v) \notin E_T$ ,  $T' = (V, E_T \cup \{(u,v)\})$ ,  $C =$  unique cycle in  $T'$ .

Hence, must exist a different edge in the cycle;  $\exists (u',v') \in C$ ,  
 $(u',v') \neq (u,v)$ ,  $u' \in V_i$ ,  $v' \notin V_i$

Must be the case that  $w(u',v') \geq w(u,v)$  (by min definition)

Then  $T^* = (V, E_{T^*})$   $E_{T^*} = (E_T \setminus \{(u',v')\}) \cup \{(u,v)\}$

$T^*$  acyclic since  $C$  is an unique cycle in  $T'$

$T^*$  connected (travel along other part of the cycle)

$\Rightarrow T^*$  spanning tree.

Also:  $w(T^*) = w(T) + w(u,v) - w(u',v') \leq w(T)$

$\Rightarrow T^*$  is MST.

### Kruskal's Algorithm

(\*) always add  $(u,v) \in E$  such that  
 $w(u,v) = \min \{ w(u',v') \in E : u',v' \text{ in different CCs of } F \}$ .

$E_T = \emptyset$

For each  $v \in V$ , Make-Set( $v$ )

$O(n)$

Put  $E$  into a min heap by weight:

$O(m)$  Bold Min Heap

While  $|E_T| < n-1$ :

$(u,v) = \text{Extract-Min}(E)$   
 $x = \text{Find}(u)$ ,  $y = \text{Find}(v)$   
If  $x \neq y$ :  
 $E_T = E_T \cup \{(u,v)\}$

$m$  total operations  
 $O(m \log(m)) = O(m \log(n))$

$2m$  Find(s) }  $O(m \log^*(n))$

$$\left. \begin{array}{l}
 \text{If } x \neq y: \\
 \left\{ \begin{array}{l}
 E_T = E_T \cup \{(u,v)\} \\
 \text{Union}(x,y)
 \end{array} \right. \\
 \dots \\
 (n-1) \text{ Unions}
 \end{array} \right\} O(m \log^*(n))$$

Total:  $O(n \log(n))$

Other Algorithms:

Prms

Tarjan 95,  $O(m)$  expected time randomized algorithm

Current best:  $O(m \alpha(n))$  deterministic

?  $O(m)$  deterministic (open problem)