Lec 21 Min Spanning Tree

MST

<u>Def</u> Given undirected, connected graph $G = (V, E)$, a spanning tree is tree $G' = (V, E')$ with $E' \subseteq E$

A min spanning tree (MST) for undirected connected weighted graph $G = (V, E, w)$ is $ST = (V, E')$ of G with min weight sum for E'.

<u>Apps</u>
- Connecting things with min cost
- Approximate TSP

<u>Prop</u> Given tree:

add edge creates exactly <u>one cycle</u>

then removing any edge in this cycle creates tree again

<u>Prop</u> Light edge property

$\forall$ undirected, conn, weighted G with $|V| \geq 2$, — unique weights
$\forall$ $U \subseteq V$, $|U| \geq 1$,
the min edge e from U to $V \setminus U$ is in MST

<u>Proof</u> <u>C1</u> If e is only edge btwn U and $V \setminus U$ then duh

<u>C2</u> Else AFSOC $e \notin$ MST, then $\exists\, e' \in$ MST that goes btwn U and $V \setminus U$, $e' \neq e$, and e' forms cycles with e if e added to MST

Add e to the MST and remove e'. We still get spanning tree but costs less. ※

<u>Prop</u> Heavy edge prop

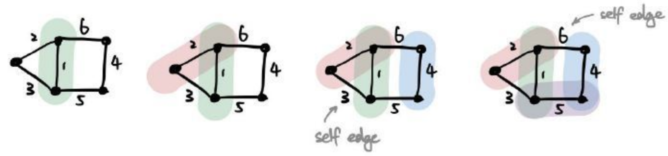
the heaviest edge in any cycle is not in the MST

MST Algs

All $O(m \lg n)$ work, span maybe different

Kruskal

sort edges by weight
for i in 0..n-1 :
  if (u,v) = E[i] not self edge :   ← check using union find
    contract (u,v) , add (u,v) to MST



Prim ← same cost analysis as Dijkstra

PFS with $p(v) = \min_{x \in X} w(x,v)$

Sleator-Tarjan ← $\exists$ lg n method to find heavy edge in cycle

for $e = (u,v) \in E$:
  add e to MST
  if new cycle formed :
    remove heaviest from that cycle

Borůvka (1926 ish), parallel

boruvka (G = (V, E, w)) =
  if |E| then $\emptyset$
  else
    for every vert find min weight e
    add e to MST
    G' = contract all edges identified
  in
    recur on G'
  end

Cost :

Every step reduce by at least $\frac{1}{2}$
so worse case lg n steps

	W	S
per iter: Find min	m	lg n
per iter: Contract	m	$\lg^2 n$
Total	m lg n	$\lg^3 n$

$\lg^2 n$ possible using star contraction