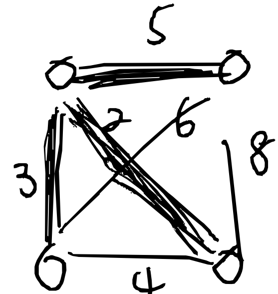
Minimum Spanning Trees.

A spanning tree T of an undirected graph is a subgraph that is a tree and contains all vertices.

Minimum spanning tree (MST) of a weighted undirected graph is a spanning tree with min total edge weights.

Problem: compute MST



(m = # edges, n = # vertices)

Kruskal: add edges to tree T in inc order by weights as long as no cycles $O(m \log m)$ time

Prim: grow T from a single vertex, add the vertex closest to T each time. $O(m \log n)$ time.

Today & next: better understanding of MST, and faster algo

Assume all edge weights are distinct

Q: Is MST unique? (YES)

How do we tell if an edge is in MST or not?

- (u,v) is in MST if $\exists$ cut (X,Y), (u,v) is the min weight edge between X and Y.
- (u,v) is not in MST if $\exists$ a cycle s.t. (u,v) is the max weight edge.

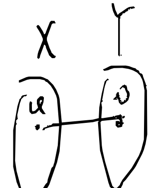
A general MST algorithm framework

algo builds the MST T by coloring edges:

- blue edge: this edge must be in T
- red edge: no longer consider

partition of the vertex set

If "discover" a cut (X,Y) s.t. (u,v) is the min weight edge between S and T,

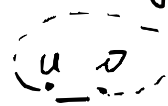


color (u,v) blue

algo may not explicitly examine all edges in the cut,

If "discover" a cycle s.t. (u,v) is the max weight edge, color (u,v) red.

may not explicitly examine all edges!



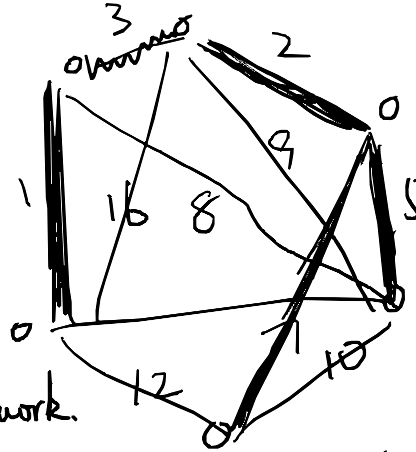
Keep coloring edges (in some order)

- blue edges form a set of trees (subtrees of MST)
- until all blue edges form one single tree (MST).

Kruskal: color edges in the inc order by weight.

Prim: color all edges between the newly added vertex and T.

Boruvka: for each blue tree, choose the min weight outgoing edge. Color all of them blue simultaneously. 1 round. Repeat until there's 1 blue tree.



Correctness of the framework.

Lemma 1: No edge can be colored both blue and red.

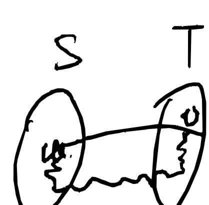
Lemma 2: Let T be MST. All $e \in T$ can be colored blue. All $e \notin T$ can be colored red.

Proof of Lemma 1: Suppose such edge (u, v) exists.

(u, v) is the min weight edge between some cut (X, Y)

(u, v) is the max weight ... in some cycle.

the cycle must cross (X, Y)



at some other edge (x, y)

$w(u, v) < w(x, y)$ but

$w(u, v) > w(x, y)$.

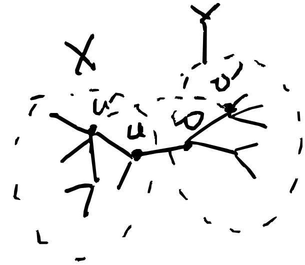
Contradiction. $\square$

Proof of Lemma 2:

Consider $(u,v) \in T$.

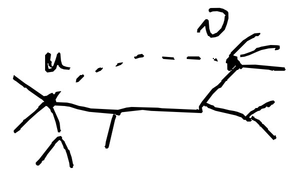
edge (u,v) connects two parts

X and Y of T.



(u,v) is the min weight edge between cut (X,Y)

o.w. replacing (u,v) with a cheaper (u',v') reduces the weight of T.



Consider $(u,v) \notin T$

(u,v) + path $u \to v$ in T forms a cycle.

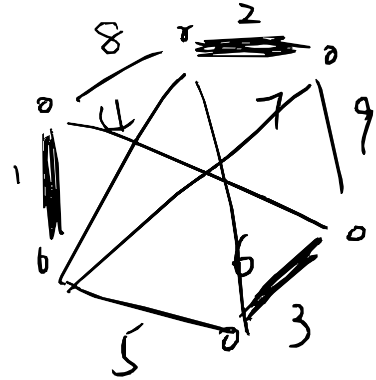
(u,v) must have max weight.

o.w. replacing a larger weight edge on this path with (u,v) reduces the weight of T. $\square$

Boruvka with "cleanup".

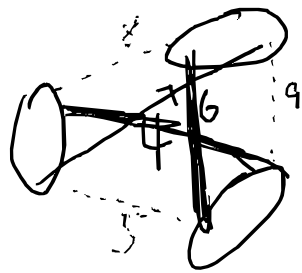
- after each round, contract each blue tree and remove redundant edges.

- solves MST for planar graphs in $O(n)$ time.



- each round reduces n to $\leq n/2$

- after contraction, the graph remains planar.



- running time: $O(n + n/2 + n/4 + \dots) = O(n)$.