Minimum spanning trees

Note Title 12/3/2012

Announcements

Dfn: Given a wieghted graph, find a tree T such that every vertex is in T and

$$\sum_{\{u,v\} \in T} w(\{u,v\}) = w(T)$$

is minimized.

Such a tree is a minimum spanning tree.

Question:

Why won't BFS/DFS work?

don't pay attention to weights

Why not shortest path tree?

saw example last

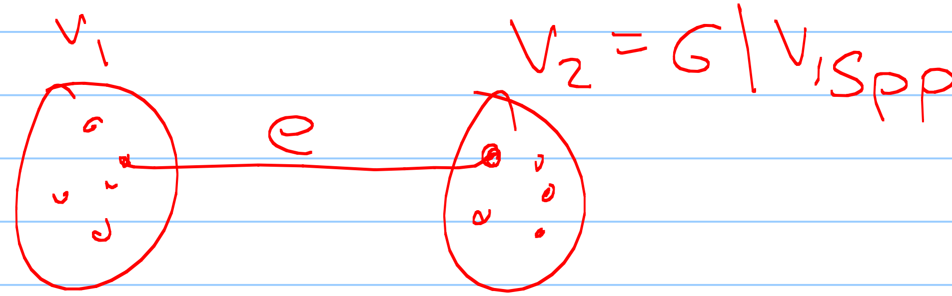
Key Fact

Let G be weighted, connected graph, & let V_1 & V_2 be a partition of V into non-empty sets.

Let e be minimum weight edge between V_1 & V_2.

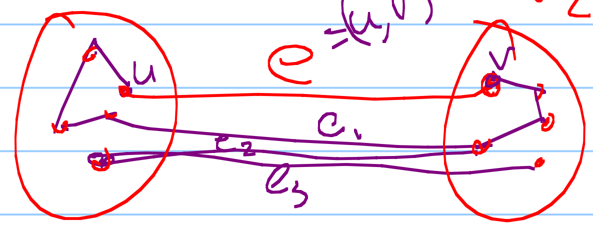
Then there is a MST containing e.

pf:



Pf by contradiction: Spps we have MST T, w/out e in it.

V_1 $\quad V_2 = G \setminus V_1$ Spp



e is smallest edge from V_1 to V_2

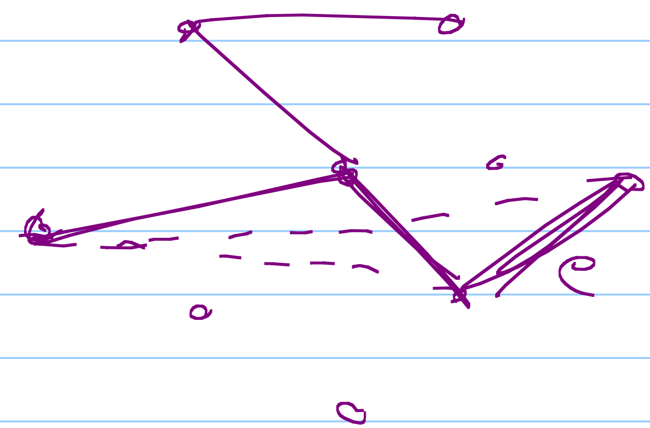
Since T is a spanning tree, must have a u to v path.

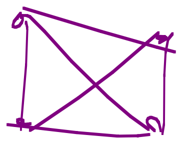
Some edge e' on path must go from V_1 to V_2, & $w(e') \geq w(e)$, since e is min from V_1 to V_2.

Make $T' = T - e' + e$. can still reach everything

So how to use this fact?

Know shortest edge in G must be in MST.





n vertices

m edges $= \binom{n}{2} = O(n^2)$

Kruskal's algorithm

Sort!

$m \log m$

Build MST in "clusters":

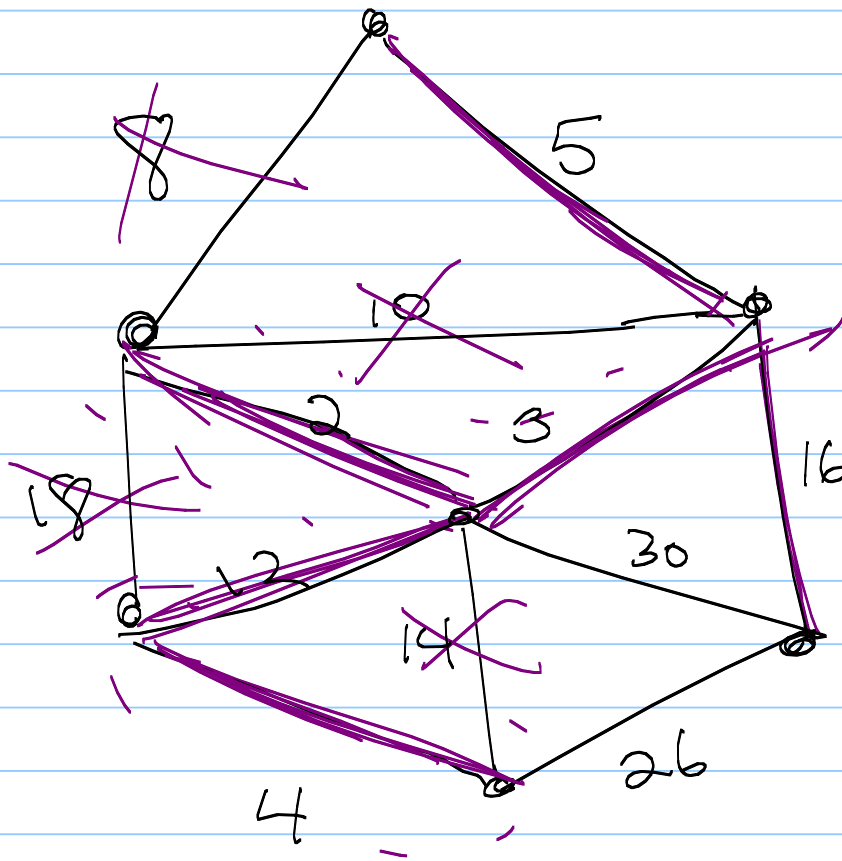
Initially, each ~~edge~~ vertex is by itself.

In a loop, take next smallest edge.

- if e connects two different clusters, add it to MST
- if e goes between 2 vertices of same cluster, discard it.

DFS (BFS: $O(m+n)$)

Ex:



Why does it work?

Relies on prev lemma:

Make V_1 my current cluster:

if $e=(u,v)$ is being added,

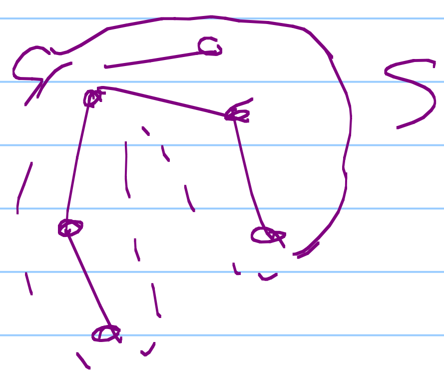
take u's cluster $= V_1$ + $G - V_1 = V_2$

e must be in MST

Another: Prim's algorithm

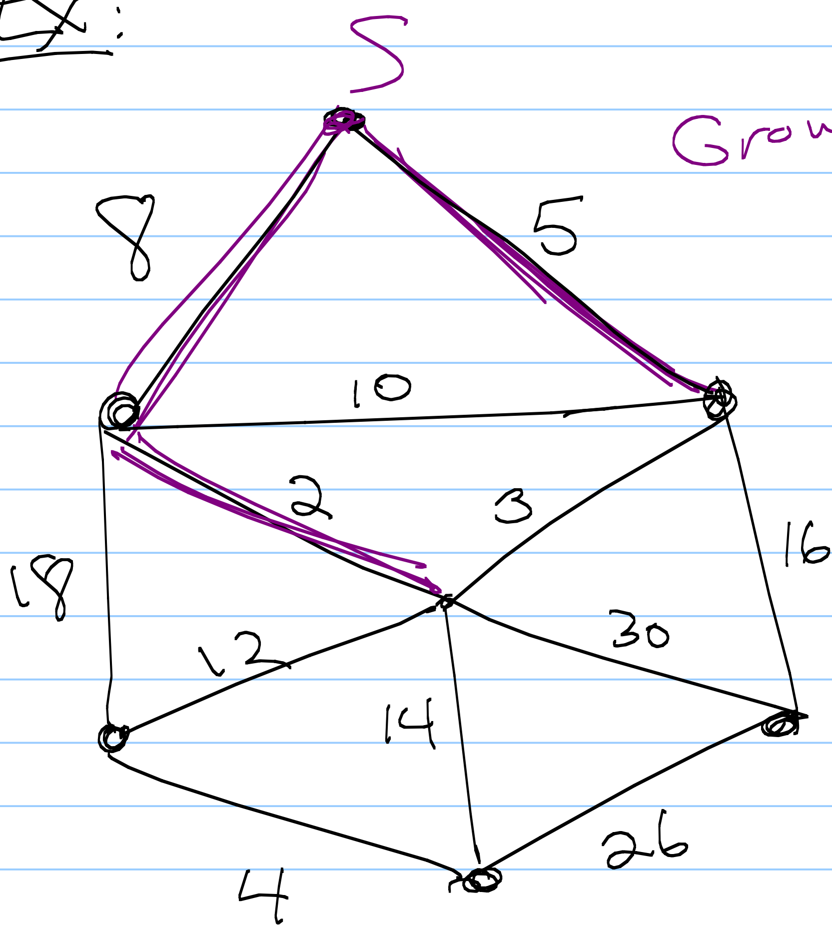
Grow MST starting from a vertex.
(similar to Dijkstra shortest path tree)

Keep a set of "reached" vertices.
At each step, add lowest weight edge going from a vertex in the set to a vertex outside.



~~take~~ min edge leaving S

Ex:



Grow as you go

Running time: (of ~~Prim's~~ Kruskal)

In a loop, take next smallest edge.

- if e connects two different clusters, add it to MST

- if e goes between 2 vertices of same cluster discard it

$O(m+n) \cdot O(m) = O(m^2)$

Faster way than DFS / BFS: growing forests using Union-Find

$O(n \log n) + \boxed{O(m \log m)}$