Minimum spanning trees

Note Title 12/3/2012

Announcements

- Lab on Thursday
- Last HW will go up today
  (not to be graded)

  one question will be on final

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.

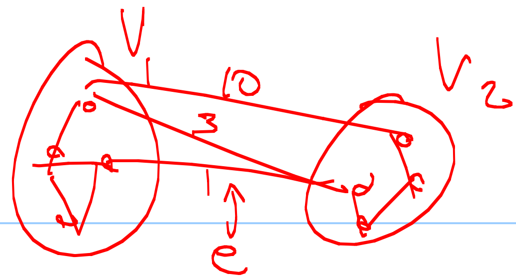
Question:

Why won't BFS/DFS work?

these don't pay attention to weights

Why not shortest path tree?

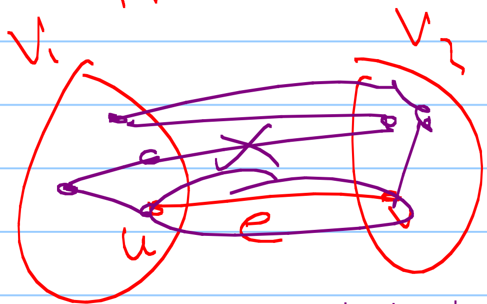
easy counterexample



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.

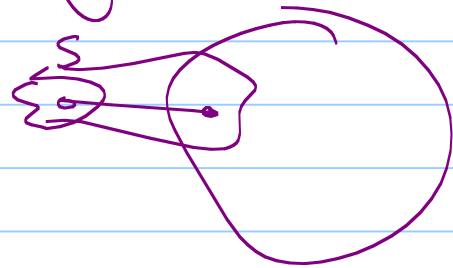
pf: Suppose e is not in MST



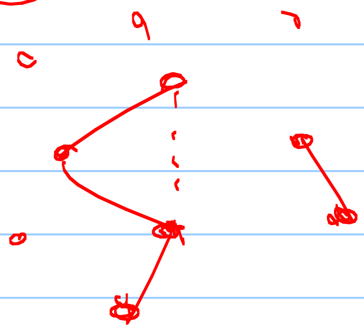
MST T must have u-v path which has at least one edge b/t V_1 & V_2.
delete that edge & replace w/ e

So how to use this fact?

1. start w/ some vertex v
   put $S = \{v\}$
   At each stage, add min edge b/t S and $V-S$



2. Know min edge is in MST



Kruskal's algorithm

Build MST in "clusters":

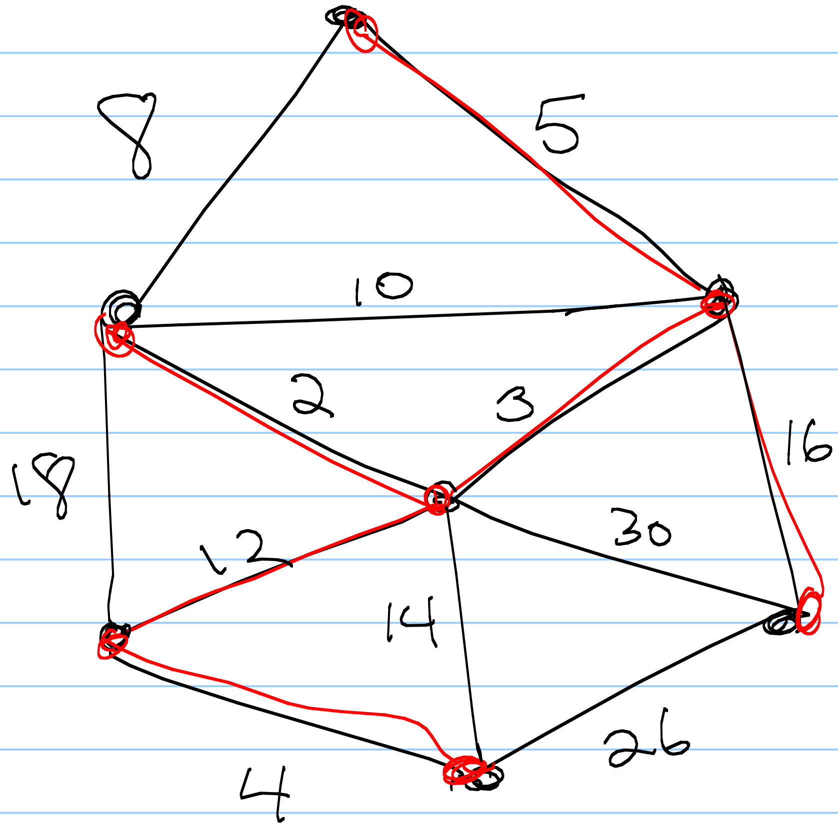
Initially, each edge is by itself.

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.

Implement: Union-Find data structure

Ex:



Why does it work?

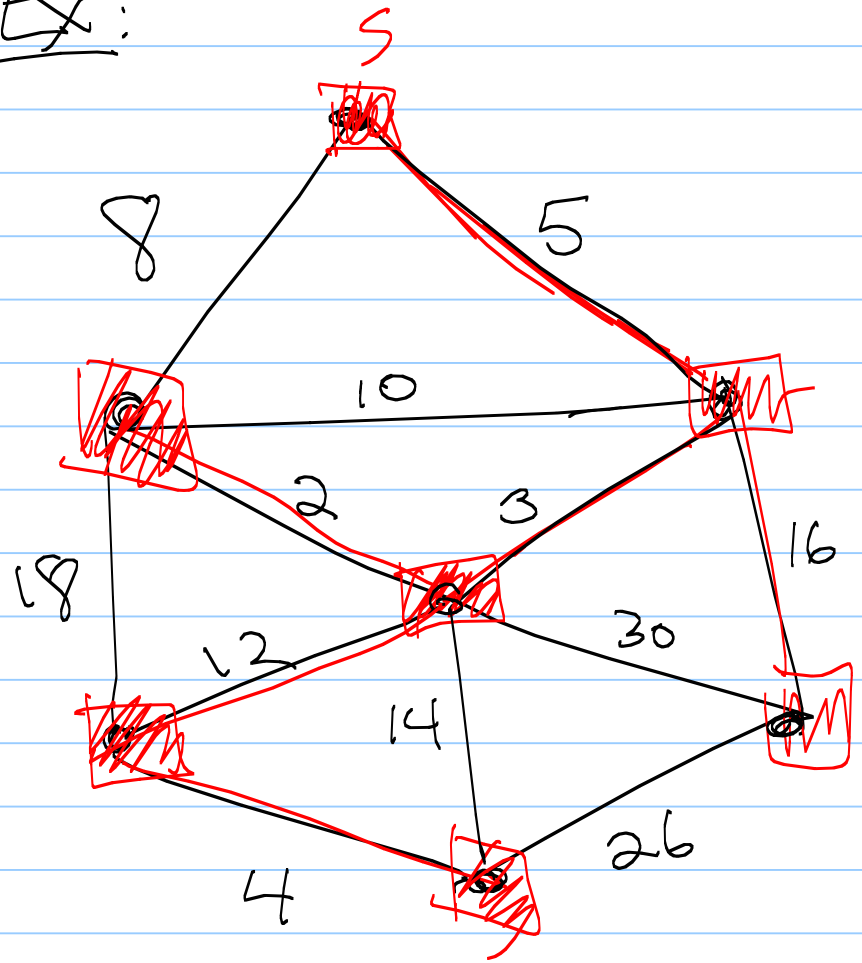
Another: Prim's algorithm

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.

Why? use fact:
let V_1 = set S
$V_2 = V - S$
& take min edge b/t partition

Ex:



$$m \le n^2$$
$$\log m \le \log n^2 = 2 \log n$$

Running time: (of Prim's)

Variant of Shortest path tree alg.

Use priority queue

$$\sum_v d(v) \cdot \log n + m \cdot \log n$$
$$= O\big((n+m) \log n\big)$$
$$= O(m \log n)$$