CSE 102 11-16-21

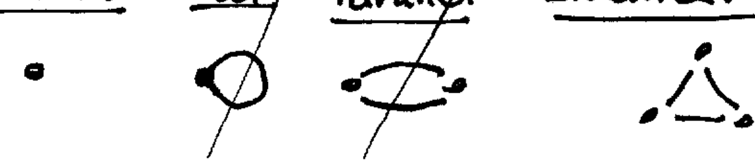
- hw 6 due tonight ✓
- Programming Project due Monday —
- midterm exam 2 Thursday (solns tonight) ✓

Minimum Weight Spanning trees

(CLRS chap 23)

- Read handout on graph theory.
- G is connected iff for all $u, v \in V(G)$, G contains a u-v Path.
- A cycle is a non-trivial closed Path.

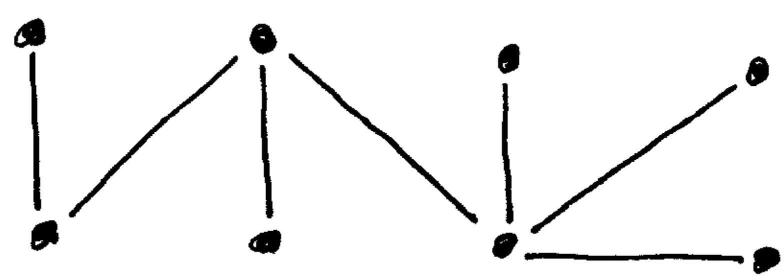
trivial ~~loop~~ ~~Parallel~~ smallest cycle



- G is Acyclic (or a forest) iff it contains no cycles.
- A tree is a subgraph that is both acyclic and connected.

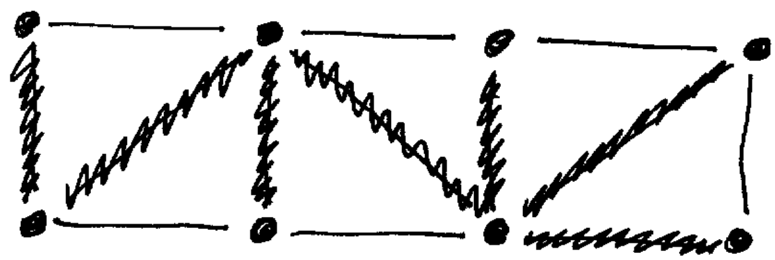
Read Treeness Theorem in Graph theory handout.

Ex.



- A subgraph of G is a <u>spanning tree</u> iff it is (1) a tree and (2) includes all vertices of G.

<u>Ex.</u>

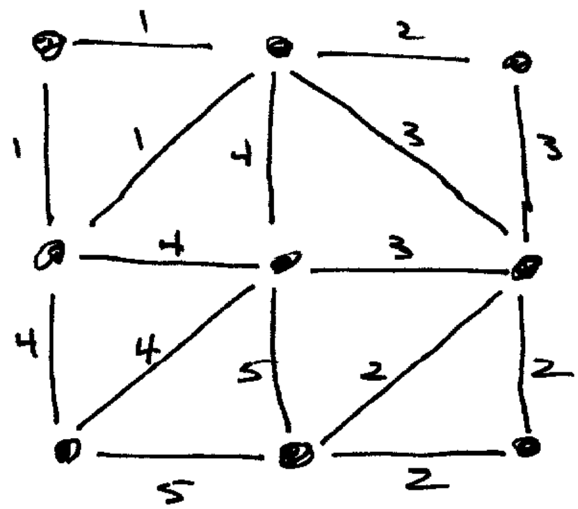


Suppose $G = (V, E)$ is a weighted graph, i.e. we have

$$w : E \longrightarrow \mathbb{R}$$

<u>Theorem</u>: a graph G contains a sp. tree iff it is connected.

Ex.



- The <u>weight</u> of a sp. tree is the sum of weights of its edges

$$\sum_{e \in T} w(e)$$

<u>Problem</u> (MWST)

determine a spanning tree of G with minimum weight.

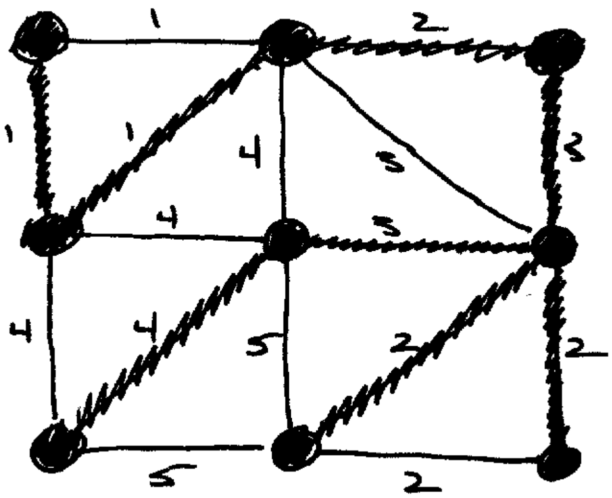
Exercise

Prove that if a weighted connected graph has distinct edge weights, then it has a unique minimum weight sp. tree.

Prim's Algorithm (sec. 23.2 in CLRS)

- choose an initial vertex (which is a tree)
- amongst all edges incident with the current tree whose addition would not create a cycle, choose one of minimum weight, add it to tree.
- stop when $n-1$ edges have been selected. (where $n = |V(G)|$.)

Ex.



$w(T) = 18$

note: Prim mantains a tree which becomes a sp. tree on last step.

Theorem

This sp. tree has minimum possible weight.

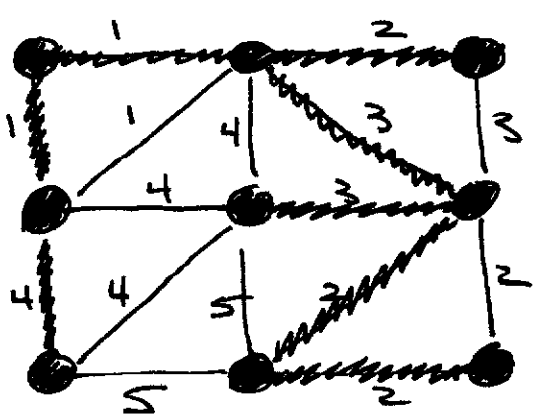
note: Implementation uses a priority queue.

Kruskal's Algorithm (sec. 23.2)

- choose an edge of min weight (which is a forest, including all vertices.)
- amongst all edges which do not create a cycle with the current forest, choose one of minimum weight and add it.
- stop when $n-1$ edges have been added.

at each stage, Kruskal has created a forest with n vertices & selected edges.

Ex



Theorem

This sp. tree that Kruskal creates is a MWST.

Proof.

Let T be the sp. in G created by Kruskal, let S be any other sp. tree in G (i.e. $S \neq T$). we must show

$$w(T) \leq w(S).$$

Let $e_1, e_2, e_3, \ldots, e_{n-1}$ be the edges of T in the order selected by Kruskal. Since $S \neq T$ there is a 1st edge e_k <u>not</u> in S, i.e.

$$\{e_1, \ldots, e_{k-1}\} \subseteq E(S)$$

$$e_k \notin E(S).$$

Let H be the subgraph

$$H = S + e_k$$

By the treeness thm, H contains a unique cycle, call it C.

Note: C must contain an edge e of S which is not in T, for otherwise C is contained in the acyclic T.

Now Remove e from H, to obtain

$$R = H - e = S + e_k - e$$

Since R is conn. & has $n-1$ edges, it is a sp. tree.

Kruskal gurantees that

$$w(e_k) \le w(e)$$

(<u>why</u>? e does not form a cycle with $\{e_1, e_2, \ldots, e_{k-1}\} \subseteq E(S)$. Thus if $w(e) < w(e_k)$, then Kruskal would have picked e on k^{th} step instead of e_k. $\therefore w(e_k) \le w(e)$.)

Thus R is a sp. tree with one more edge in common with T than S had:

$$\{e_1, \ldots, e_{k-1}, e_k\} \subseteq E(T) \cap E(R)$$

Also $\quad w(R) \le w(S)$

If $R = T$, we're done. If $R \neq T$, do it over with R in place of S.

eventually we must reach T.

$$\begin{array}{c} w(T) \le \cdots \le w(R_2) \le \underset{\overset{\parallel}{R_1}}{w(R)} \le \underset{\overset{\parallel}{R_0}}{w(S)} \end{array}$$

Therefore $w(T) \le w(S)$, as required. ∎