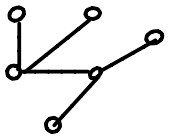
MATH F113X: Kruskal's Algorithm

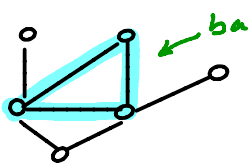
Goals:

- Understand the terms: tree, spanning tree, minimum cost spanning tree
- Understand how to use Kruskal's Algorithm to find a minimum cost spanning tree
- Know of applications of minimum cost spanning trees

1. Definitions

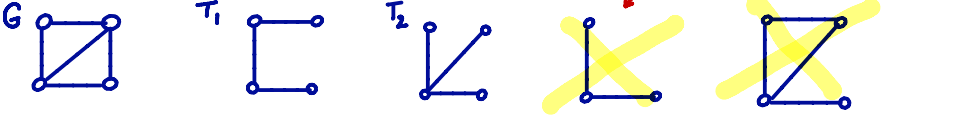
   (a) **(tree)** A **tree** is a connected graph with no circuits.

tree 



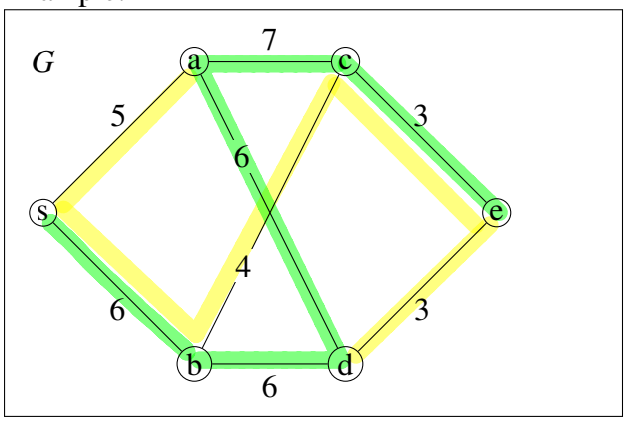
not a tree

   (b) **(spanning tree)** A **spanning tree**, T, of the graph G is a subgraph of G that is a tree and includes every vertex of G.

eg: 

   (c) **(minimum cost spanning tree)**
   Given a graph with weights or costs assigned to each edge, a minimum cost spanning tree (MCST) is a spanning tree with the smallest total edge weight (or cost).

2. Example:



T_1 has total cost:
$6+6+6+7+3 = 28$

T_2 has total cost:
$5+6+4+3+3 = 21$

MATH F113X: Kruskal's Algorithm

3. Kruskal's Algorithm

input: a graph, G, with costs (or weights) on the edges
output: a spanning tree, T, of minimum cost
Steps:

(a) (Initialization Step:) T is a graph on the vertex set of G but with no edges.

(b) (Iterative Step:)

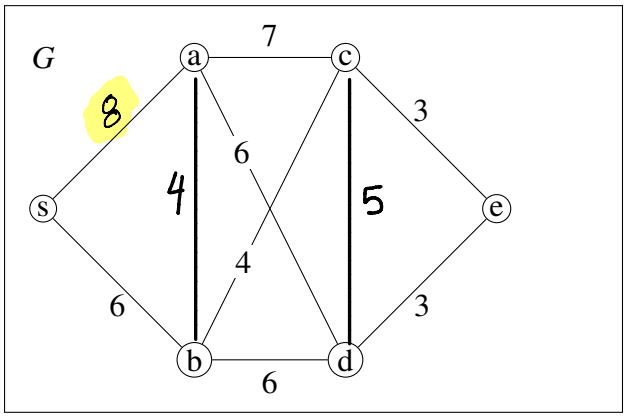
i. Select the cheapest unused edge in the graph. (Ties are broken alphabetically.)

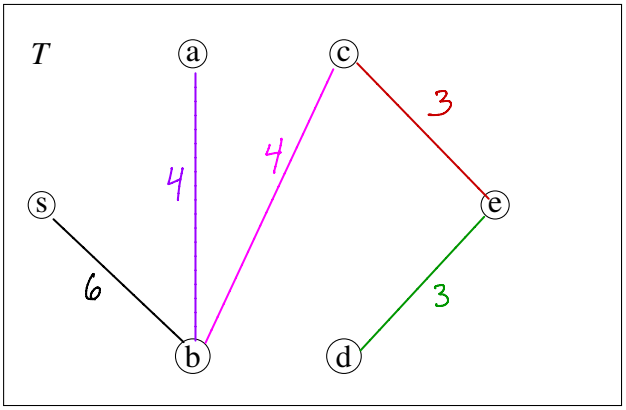
ii. If the edge does **not** create a cycle, add the edge to T. Otherwise, reject the edge.

iii. Mark the edge as used.

iv. If T is a spanning tree, terminate the algorithm. Otherwise return to the beginning of the iterative step.

4. Use Kruskal's Algorithm to find the minimum cost spanning tree for the graph G below.





total weight 20

Used?	edges	weights
(3) ✓	a b	4 ✓
	a c	7
(6) ✓	a d	6 reject
	a s	~~8~~
(4) ✓	b c	4 ✓
(7) ✓	b d	6 reject
(8) ✓	b s	6
(5) ✓	c d	5 reject
(1) ✓	c e	3 ✓
(2) ✓	d e	3 ✓

Kruskal's

5. Think of an application of ~~Dijkstra~~'s Algorithm.

Cheapest way to connect towns or computers.