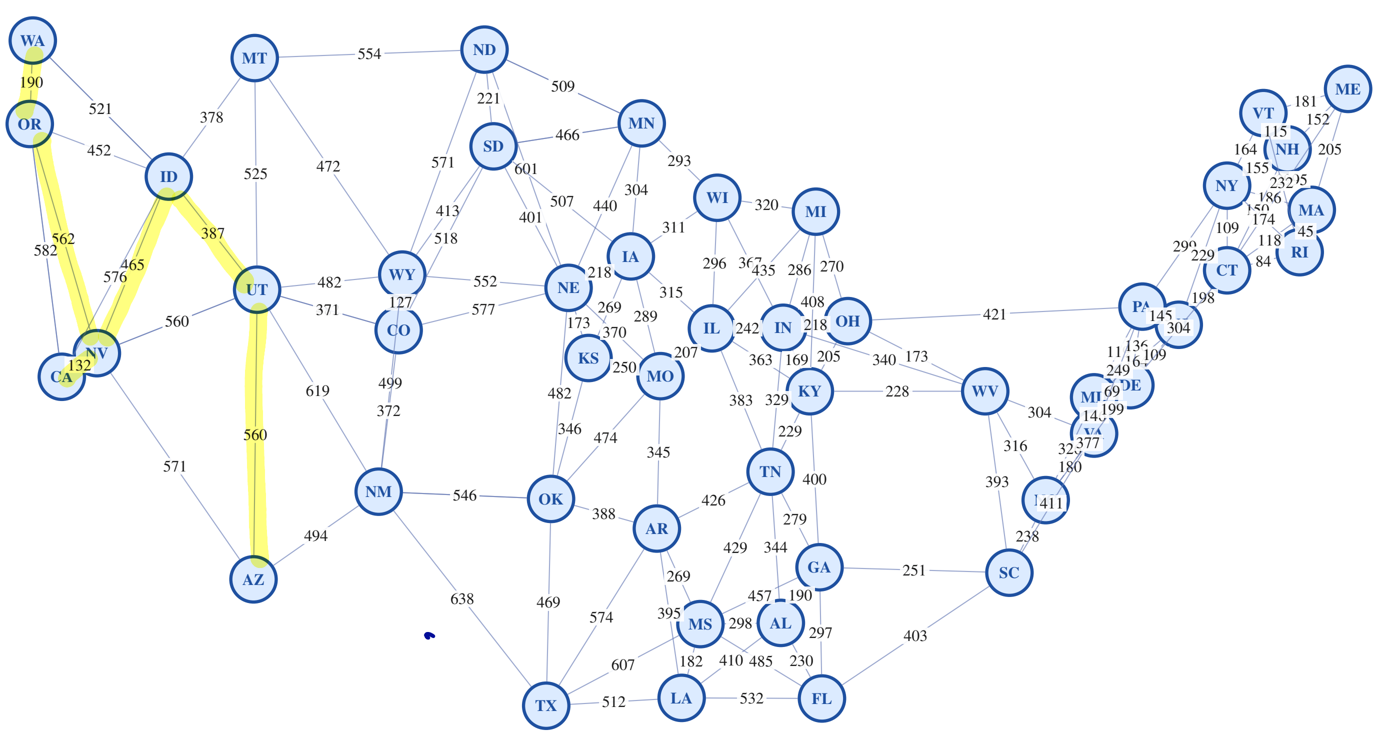
Goals:

- 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

Weighted Road Graph: 48 Contiguous U.S. State Capitals

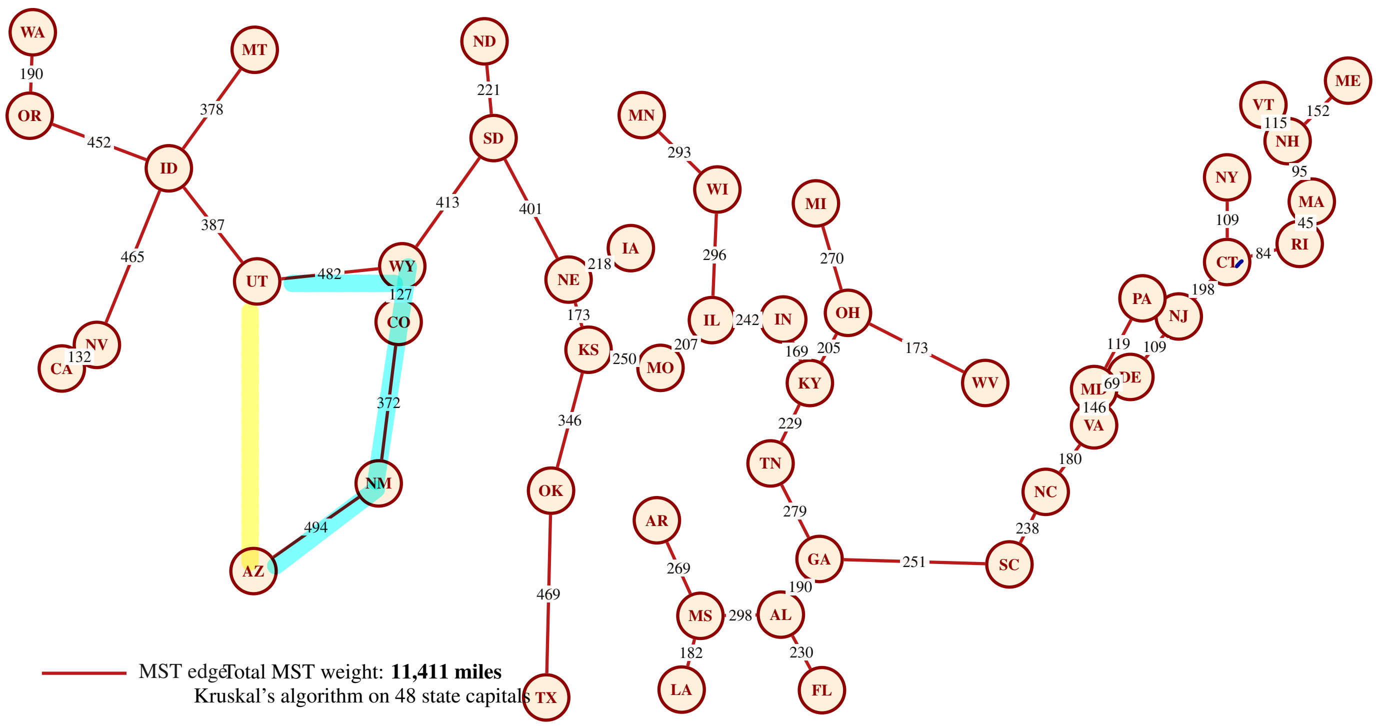
Edge weights = approximate road distance in miles. Each capital connected to its 4–5 nearest neighbors.



- Capitals of all lower 48 states
- edges indicate distance on the ground
- Imagine some catestrophic event in which all roads are damaged. Where should the goverment place its resources in order to have all capitals connected as quickly as possible?
- Start connecting close places?

Minimum Weight Spanning Tree (Kruskal’s Algorithm)

Total MST weight ≈ **11,411** miles. Red edges form the unique spanning tree of minimum total road distance.



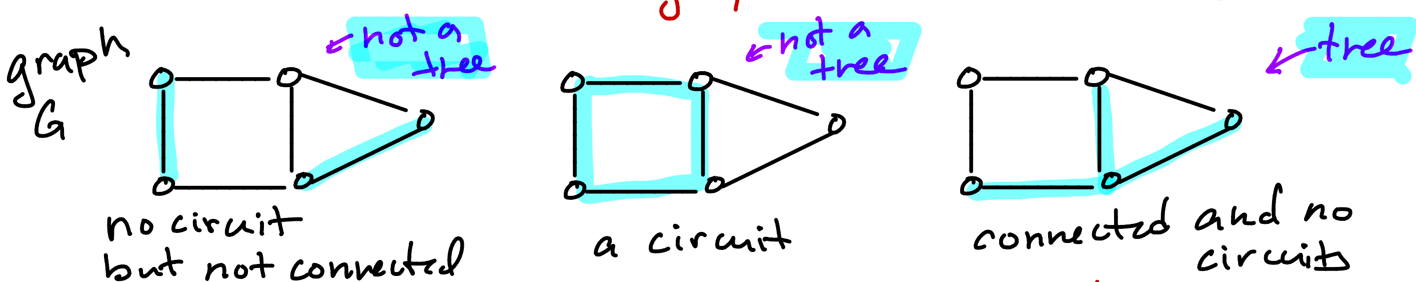
- Note, it did not choose to use the UT-AZ edge!
- Why not add UT-AZ edge?
- Why aren't there any circuits?
- Is it connected?

1. Definitions

   (a) **weighted graph** – a graph with numbers (weights) on the edges.

   The weights could represent: distance, cost, capacity, time, average daily use, ...

   (b) **tree** – A connected graph with no circuits

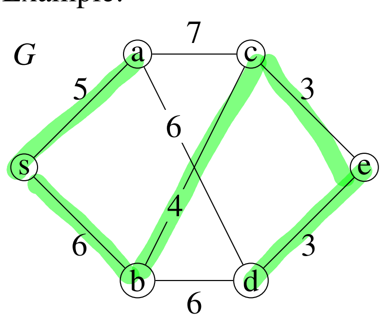


   (c) **spanning tree** – A tree in a graph that includes every vertex

   (d) **minimum cost spanning tree**

   A spanning tree in a weighted graph where the sum of the edges is as small as possible.

2. Example:



Let's just guess!

weight: 3+3+4+5+6

= 21

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.)

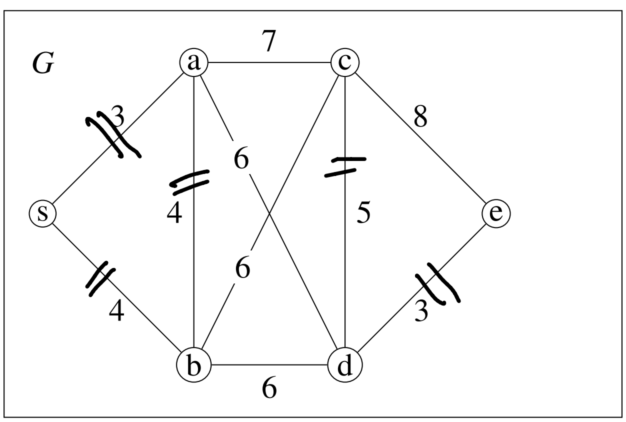
ii. If the edge does **not** create a cycle, add the edge to T. Otherwise, reject the edge.

iii. Mark the edge as used.

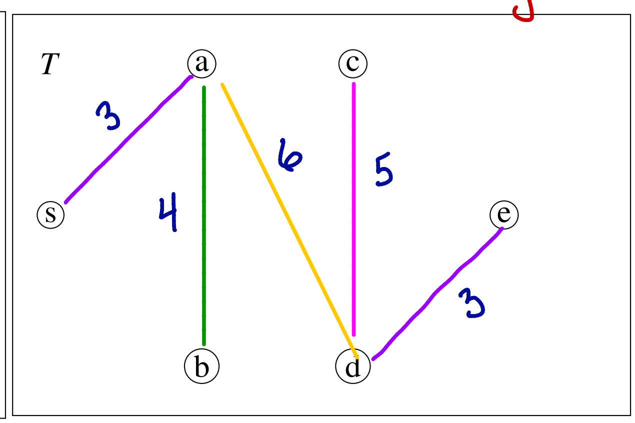
iv. If T is a spanning tree, STOP. Otherwise return to the beginning of the iterative step.

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

Given



Build our spanning tree here!



order	Used?	edges	weights (smallest first!)
2	✓	de	3
1	✓	as	3
3	✓	ab	4
	No	bs	4
4	✓	cd	5
5	✓	ad	6 stop
		bc	6
		bd	6

5. Think of an application of Kruskal's Algorithm.

vertices = webpages
edges = between two webpages if they are linked
min wgt spanning tree = fewest working links s.t. all webpages are connected.