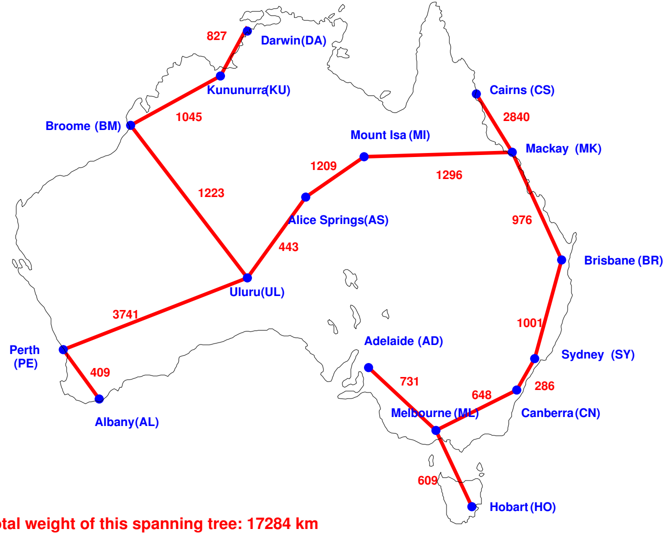
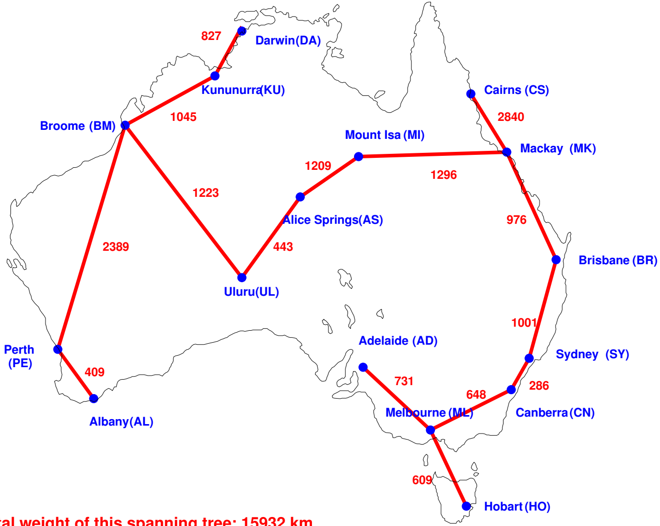
Minimum Spanning Trees

What if we have N vertices that we want to connect **as cheaply as possible?**



Total weight of this spanning tree: 17284 km



Total weight of this spanning tree: 15932 km
(cheaper!)

Minimum Spanning Trees

Suppose that we have a **weighted complete graph** with N vertices.

Minimum Spanning Trees

Suppose that we have a **weighted complete graph** with N vertices.

How can we find a spanning tree with the smallest possible weight?

Minimum Spanning Trees

Suppose that we have a **weighted complete graph** with N vertices.

How can we find a spanning tree with the smallest possible weight?

An exhaustive search of all trees is not a good idea — there are N^{N-2} spanning trees to consider (and this is a very big number!)

Minimum Spanning Trees

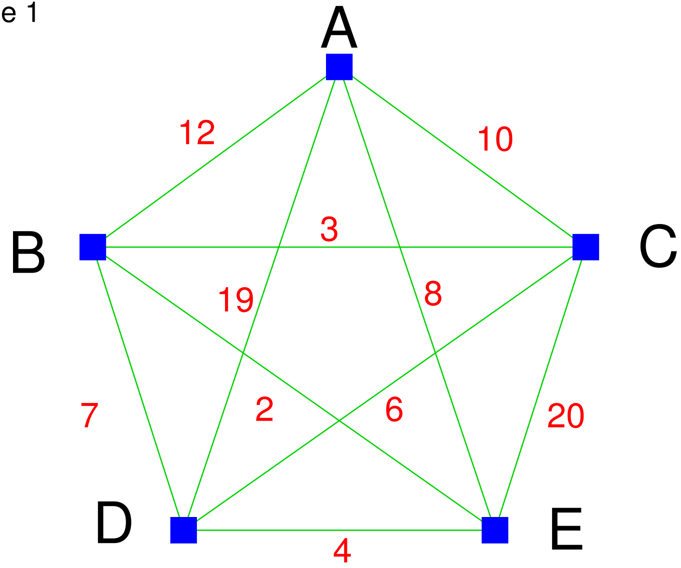
Idea (sort of like the Cheapest-Link Algorithm for finding a Hamilton circuit):

- Add edges one at a time, choosing the cheapest edge possible.

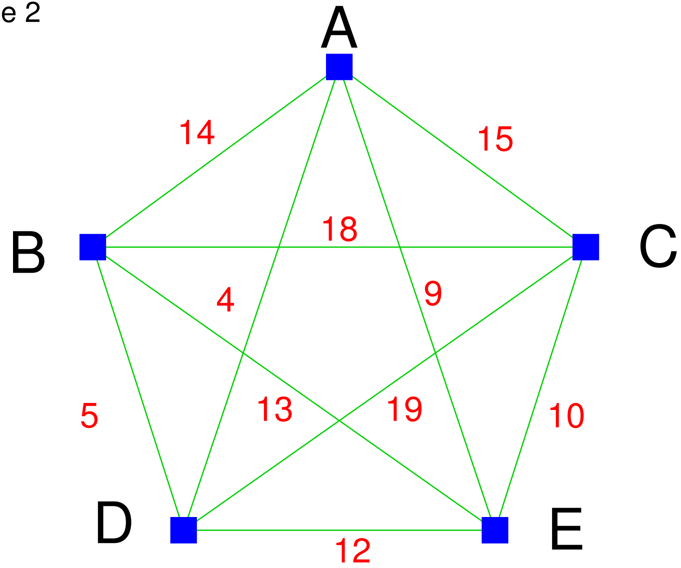
  (Break ties arbitrarily.)
- Be sure never to create a circuit.
- Stop when you have a spanning tree.

This is called **Kruskal's algorithm.**

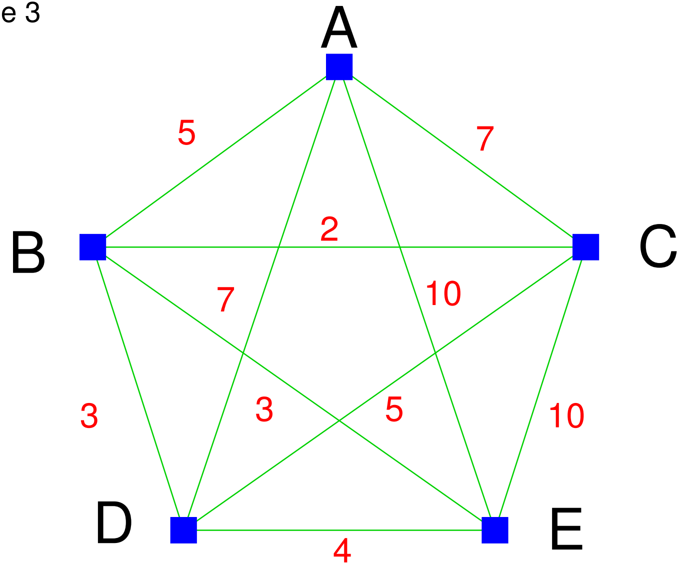
Example 1



Example 2



Example 3



Minimum Spanning Trees

And now, a miracle occurs. . .

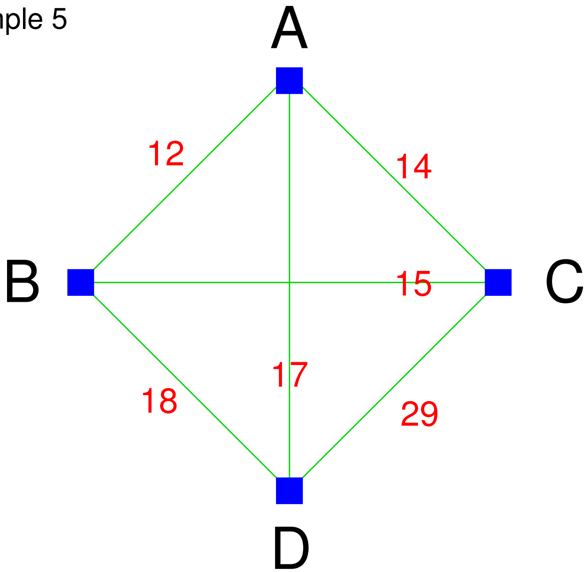
Minimum Spanning Trees

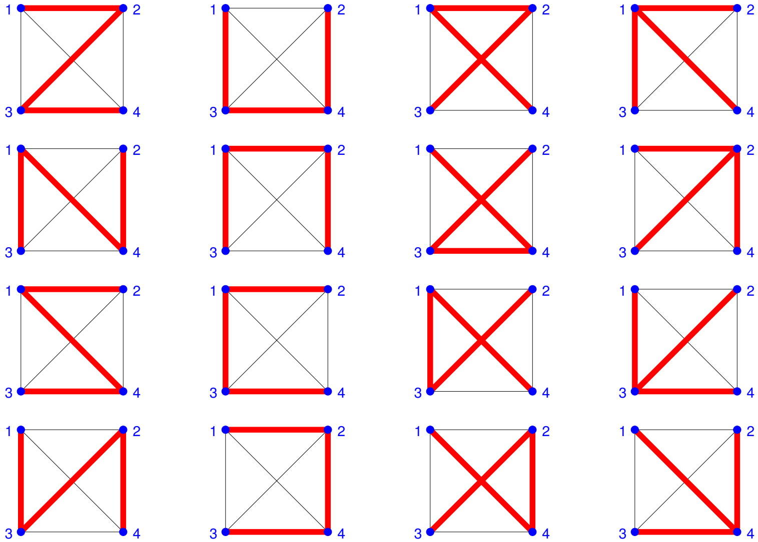
And now, a miracle occurs...

Kruskal's algorithm always works!

For example, let's look at the weighted K_4 that caused trouble for the Nearest-Neighbor and Cheapest-Link Algorithms.

Example 5




1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4

Kruskal's Algorithm

Tree	Weight	Tree	Weight
AB,AC,AD	$12+14+17=43$	AC,AD,BC	$14+17+15=46$
AB,AC,BD	$12+14+18=44$	AC,AD,BD	$14+17+18=49$
AB,AC,CD	$12+14+29=55$	AC,BC,BD	$14+15+18=47$
AB,AD,BC	$12+17+15=44$	AC,BC,CD	$14+15+29=58$
AB,AD,CD	$12+17+29=58$	AC,BD,CD	$14+18+29=61$
AB,BC,BD	$12+15+18=45$	AD,BC,BD	$17+15+18=50$
AB,BC,CD	$12+15+29=56$	AD,BC,CD	$17+15+29=61$
AB,BD,CD	$12+18+29=59$	AD,BD,CD	$17+18+29=64$