Ch. 7: Kruskal's Algorithm

Kruskal's Algorithm finds a minimal spanning tree (MST) in a weighted network (connected graph). It is similar to the Cheapest Link Algorithm. However, in this case, Kruskal's Algorithm always finds an MST, so it is <u>not</u> just an approximate algorithm.

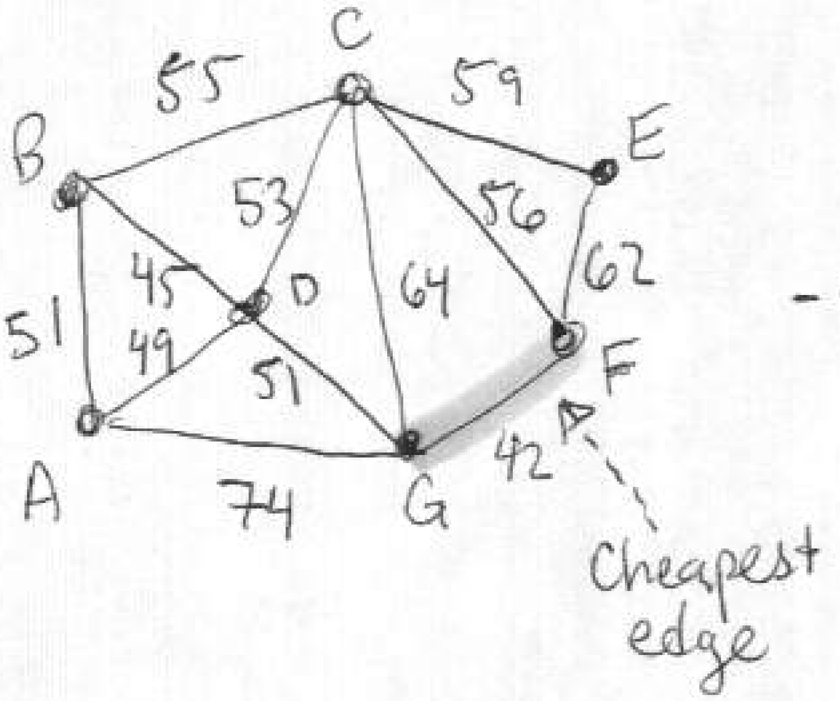
Kruskal's Algorithm

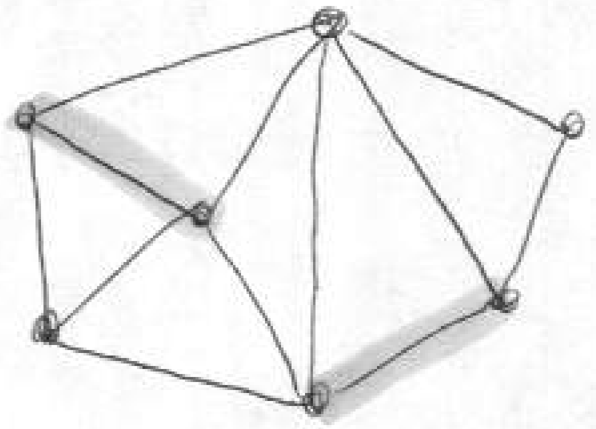
<u>Step 1</u>: Pick the cheapest edge available.
(If there is more than one, choose randomly among the cheapest options.)

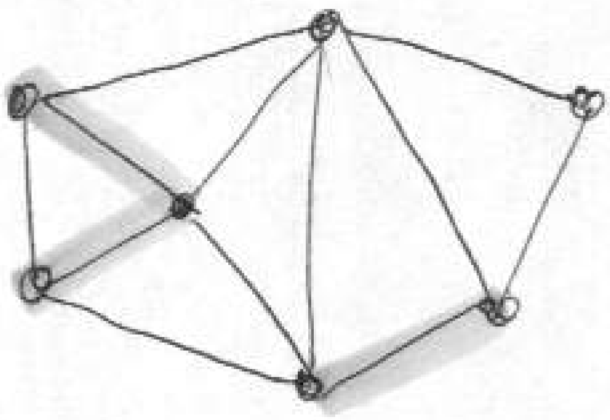
<u>Step 2</u>: Pick the next cheapest edge available.

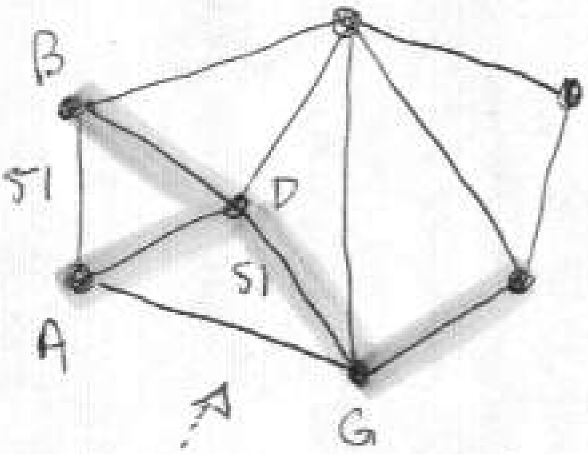
<u>Step 3</u>: Continue picking the cheapest (unchosen) edge available that does not create a circuit. After choosing N-1 such edges (where N = # vertices of the graph) then we are done.

We first apply this algorithm to the ISP example from last lecture. (This example is the one given in the textbook.)

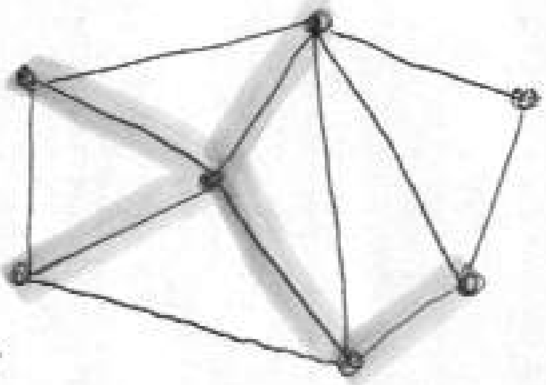


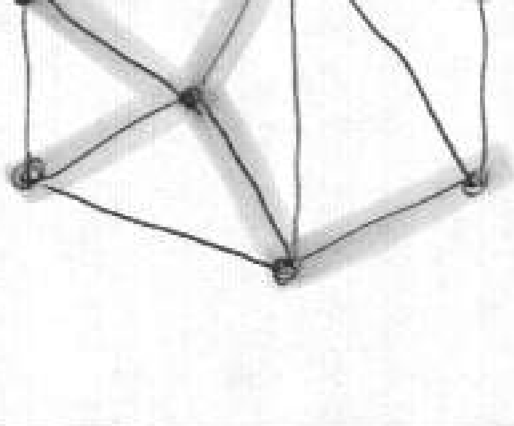






At this stage we have 2 choices: AB and DG. But AB forms a circuit, so we choose DG.



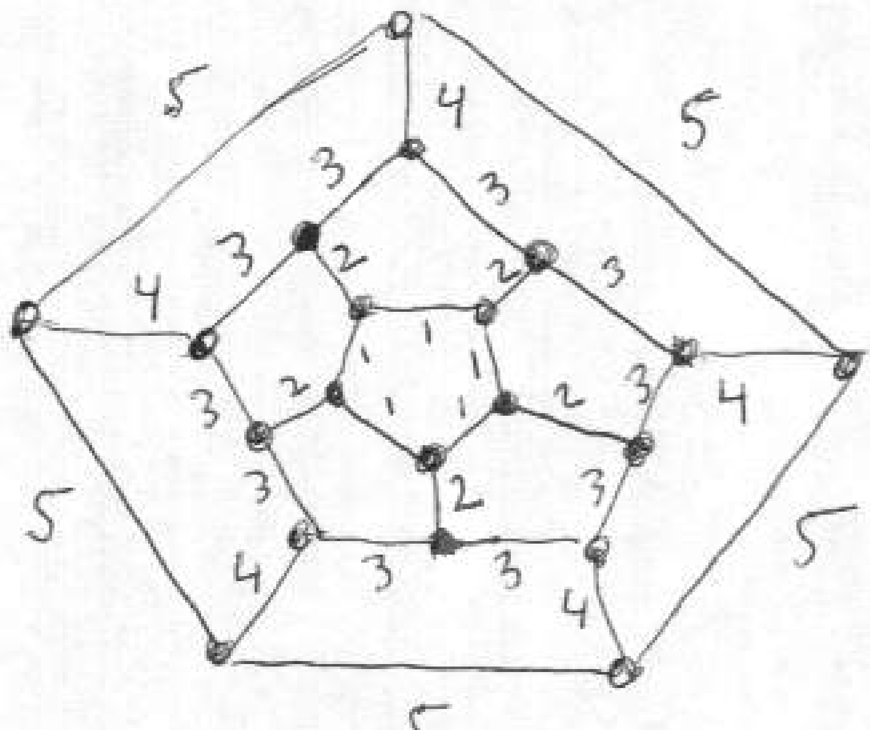


the end

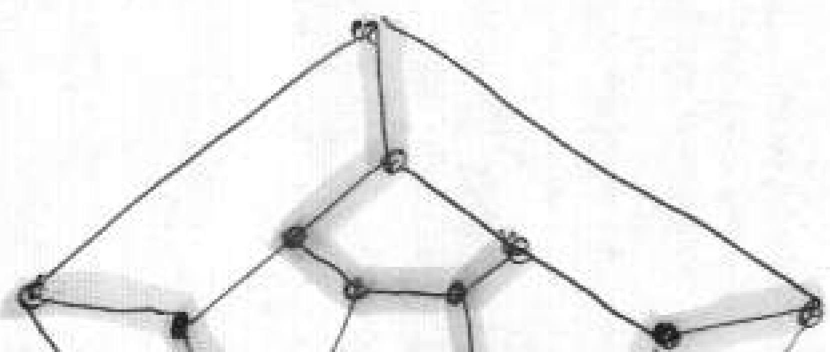
We found an MST. The total cost is

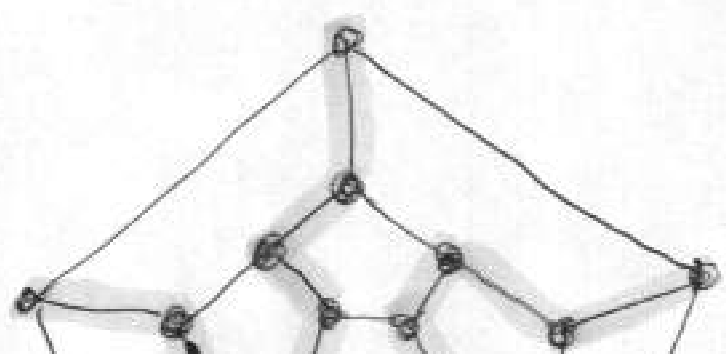
$$42 + 45 + 49 + 51 + 53 + 59 = 299.$$

Example: Let's look at Hamilton's dodecahedron graph with some weights (which represent, roughly, distance from the center):



If we apply Kruskal's Algorithm we can get different answers, depending on our choices:

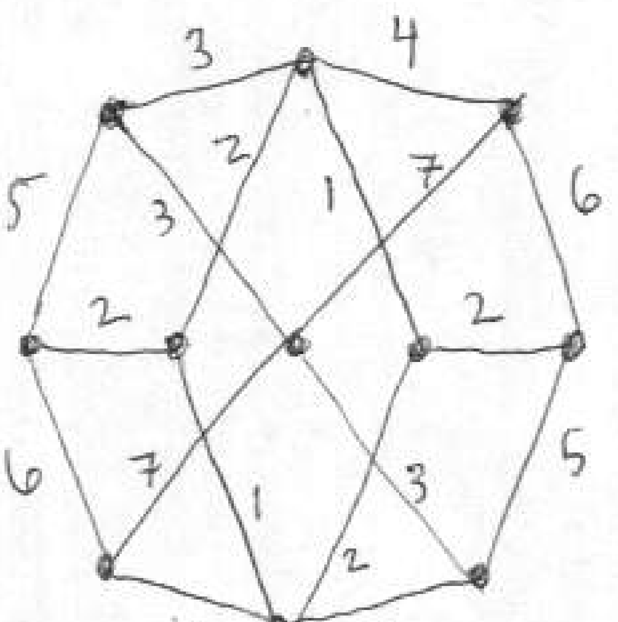




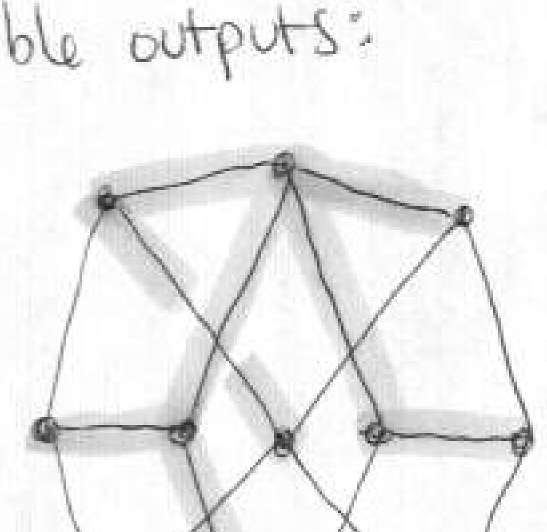
But all outputs are MST's, and they all have the same total cost:

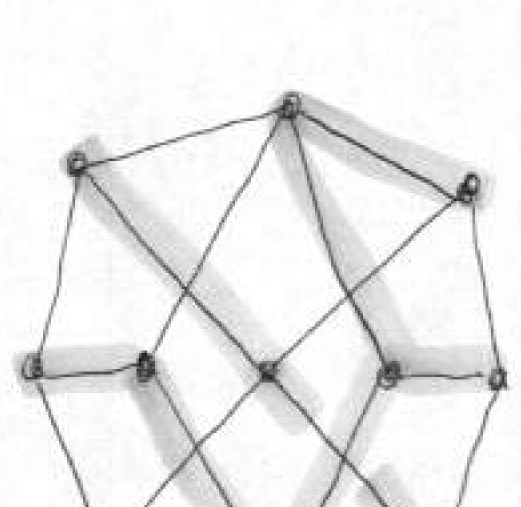
$$\text{total cost} = 1+1+1+1+2+2+2+2+2+3+3+3+3+3+4+4+4+4+4$$
$$= 4\times 1 + 5\times 2 + 5\times 3 + 5\times 4$$
$$= 4 + 10 + 15 + 20$$
$$= 49.$$

Example: Apply Kruskal's Algorithm to the following:



Two possible outputs:





As in the previous example there is more than one possible output.

Both have total cost = 25.