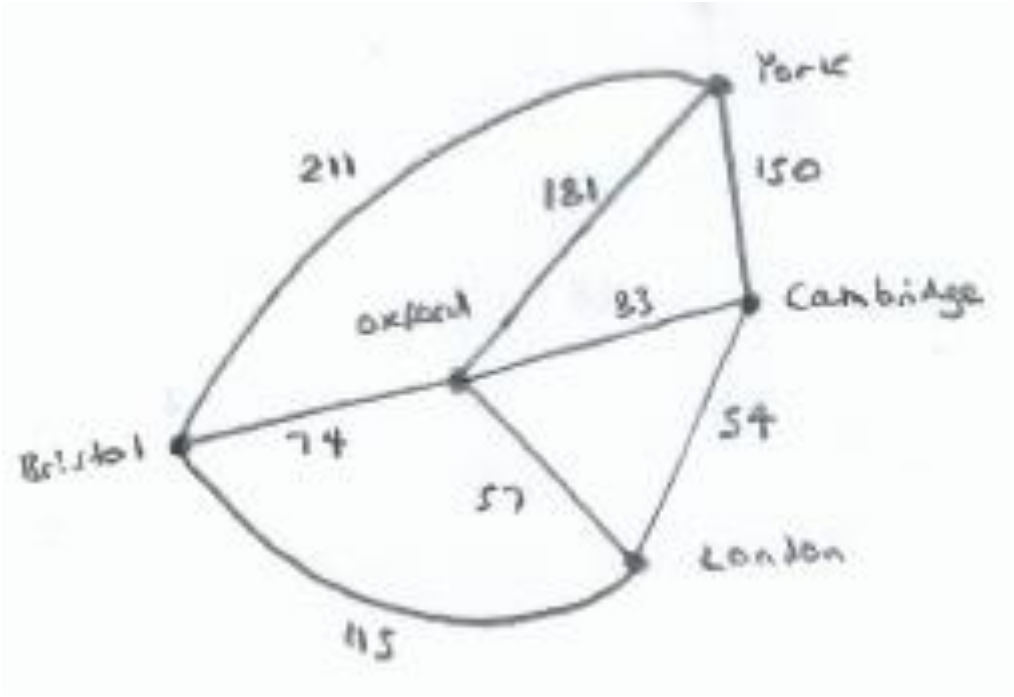
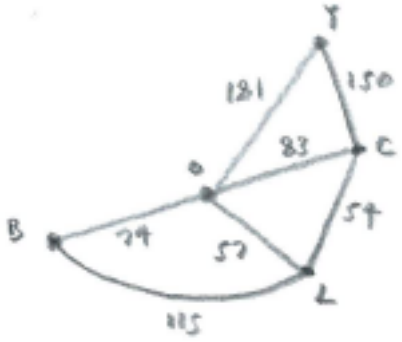
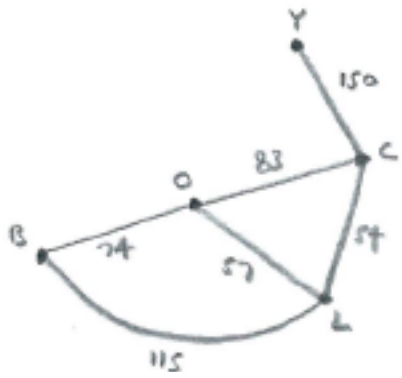
Minimum Connector Problem (2/1/2014)

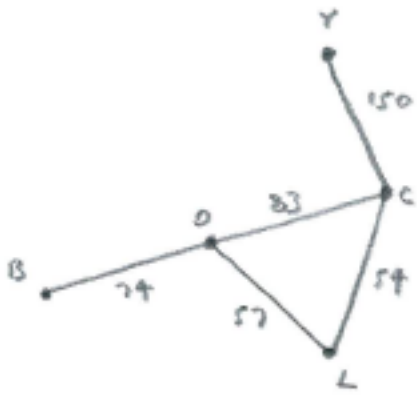
The aim is to find a minimum spanning tree; ie a tree which connects all the nodes, and which has the smallest possible total weight.



Method 1: Remove arcs in order of decreasing weight (ensuring that the network remains connected).







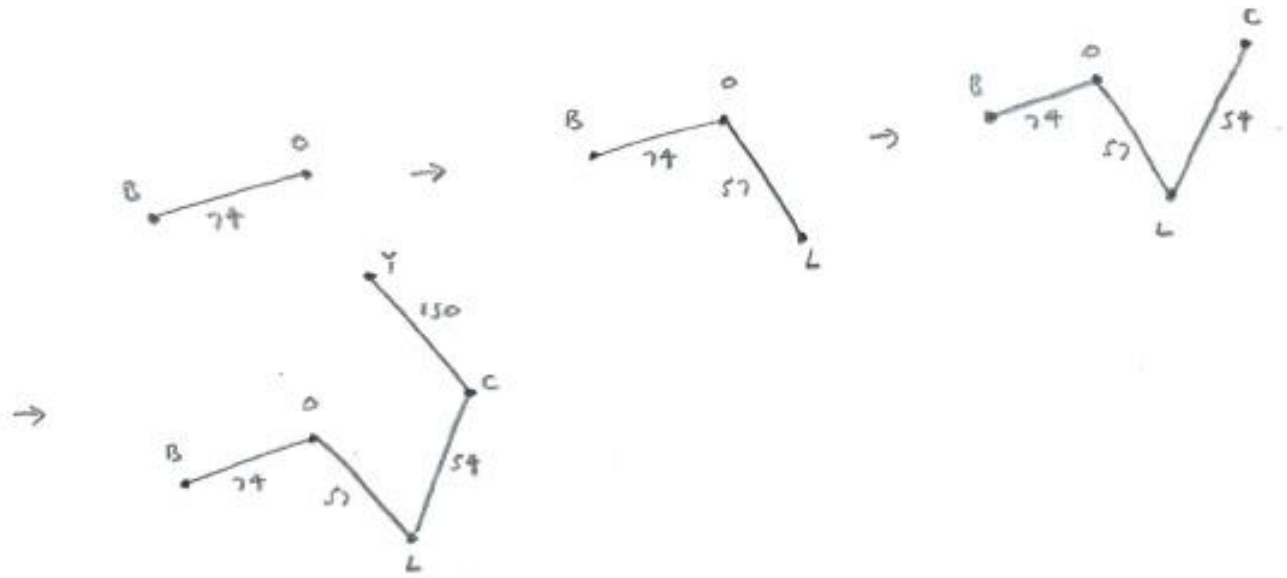
Method 2: Prim's Algorithm

(1) Start with any node.

(2) Add the arc leading to the nearest node.

(3) Add the arc leading (from either of the nodes collected so far) to the nearest new node, and repeat.

[If two nodes are the same distance from the nodes collected so far, either node may be chosen.]



[With B as the initial node, the sequence of nodes added happens to be along the path BOLCY. But had L been chosen as the initial node, for example, then the sequence would have been LCOBY.]

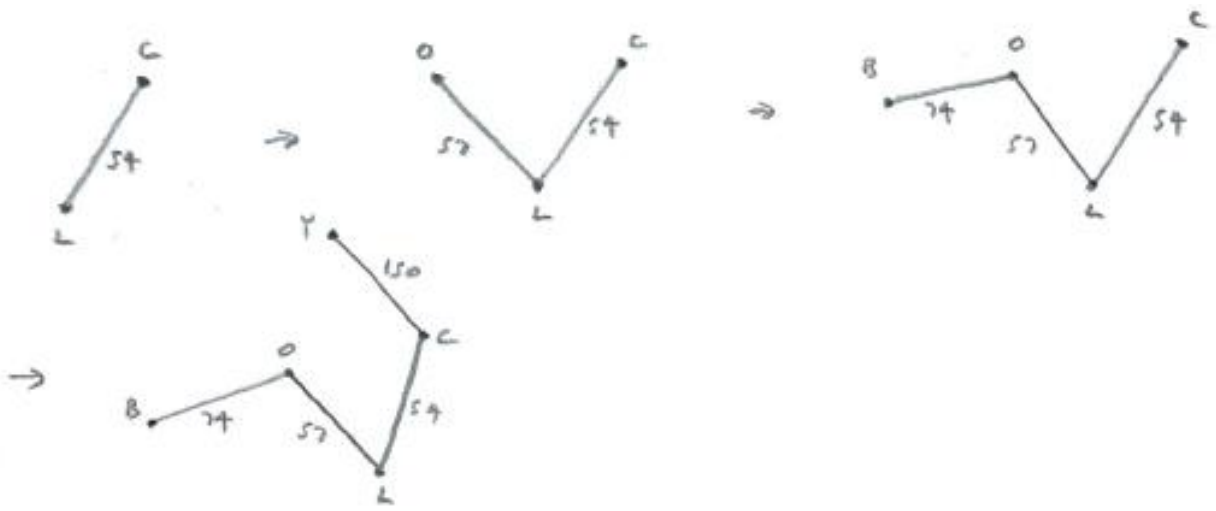
Method 3: Kruskal's Algorithm

(1) Start with the shortest arc.

(2) Choose the next shortest arc, provided it doesn't create a cycle.

[If two arcs are of equal length, then either may be chosen.]

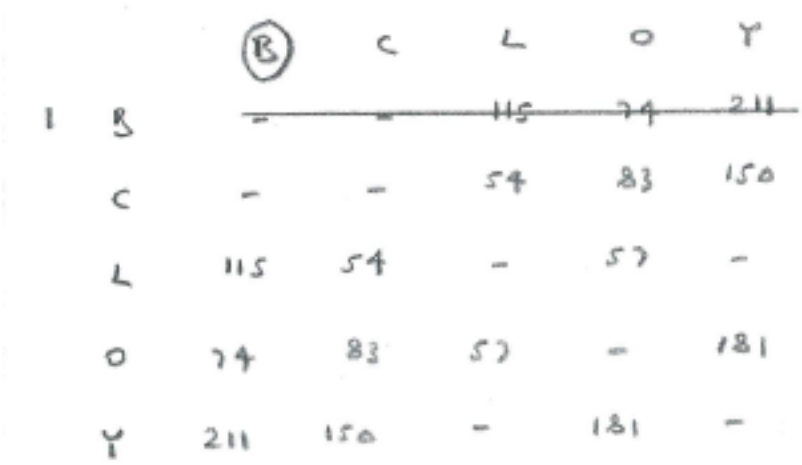
(Note: in some cases the arcs won't be connected when they are chosen - though they will eventually join up automatically.)



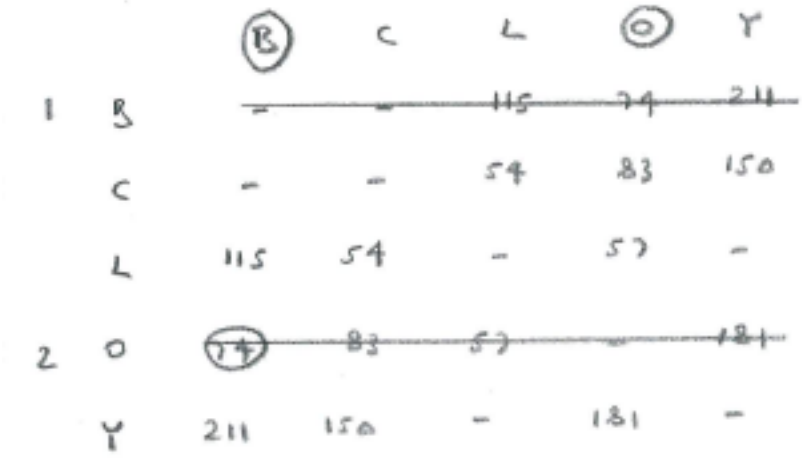
Method 4: Prim's Algorithm using the matrix of weights

	B	C	L	O	Y
B	-	-	115	74	211
C	-	-	54	83	150
L	115	54	-	57	-
O	74	83	57	-	181
Y	211	150	-	181	-

Choosing B (for example) as the initial node, we circle B in the top row (to indicate that B belongs to the nodes selected so far), and cross out the row for B (to indicate that we no longer need any arcs that end in B - as B has already been included). We can also add a column to show the order in which nodes have been added.

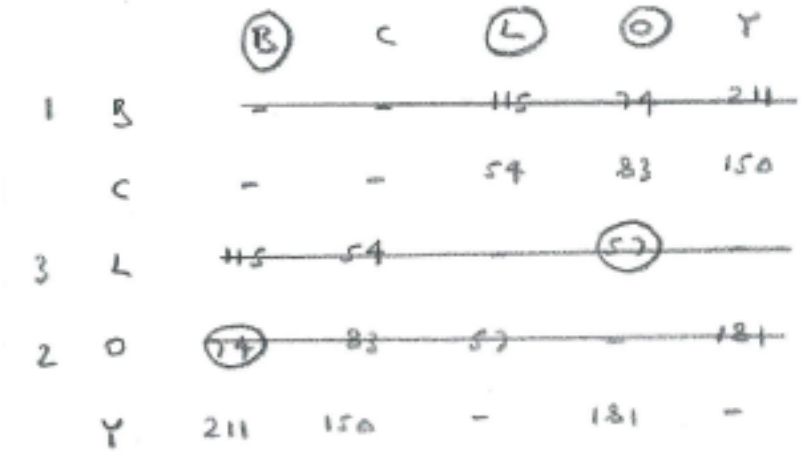


We then find the smallest weight in column B: 74 for O (indicating that O is the nearest node to B). We then circle this 74, for future reference, circle O at the top (to indicate that O belongs to the nodes selected so far), and cross out the row for O (to indicate that we no longer need any arcs that end in O).



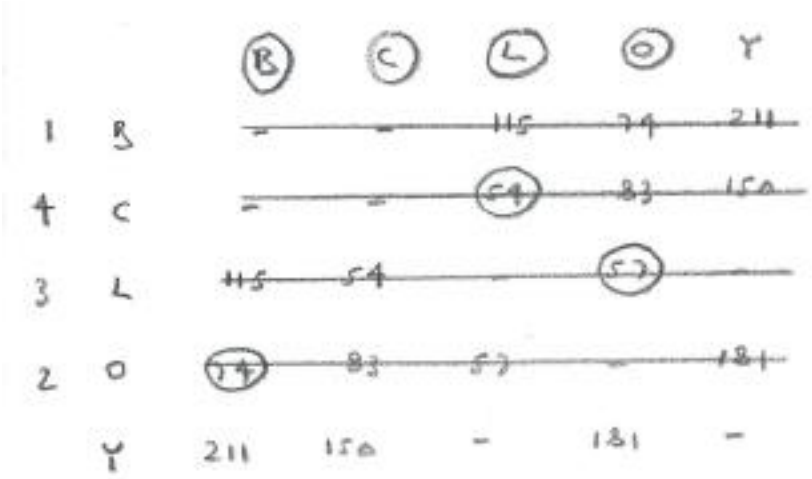
We have now created the arc BO (refer to the diagram for Method 2.)

We then find the smallest weight in columns B and O: 57 for L in column O (indicating that L is the nearest new node to O). We then circle the 57, circle L at the top, and cross out the row for L.



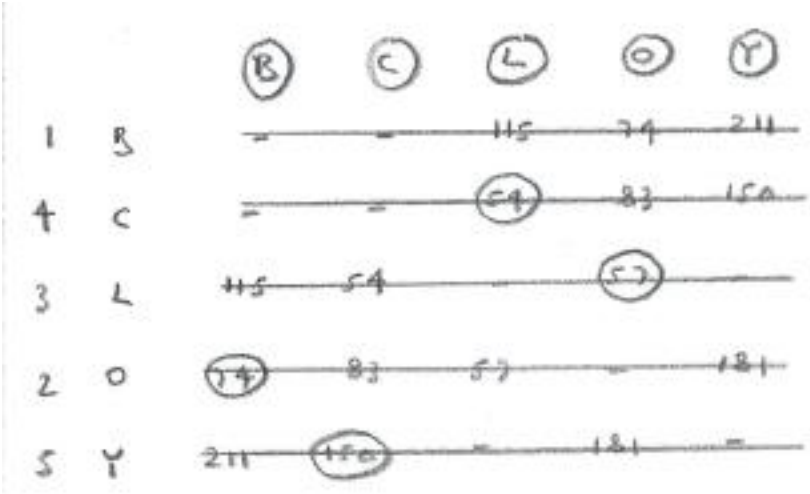
We have now created the path BOL.

We then find the smallest weight in columns B, L and O: 54 for C in column L (indicating that C is the nearest new node to L). We then circle the 54, circle C at the top, and cross out the row for C.



We have now created the path BOLC.

This just leaves the arc CY to be added:



We have now created the minimum spanning tree BOLCY.

Notes

(i) Having to check for cycles makes Kruskal's algorithm less suitable for a computer than Prim's. The matrix form of Prim's algorithm is very suited to use with a computer.

(ii) There may be more than one solution (ie with the same minimum weight), if some of the arcs have the same weight.