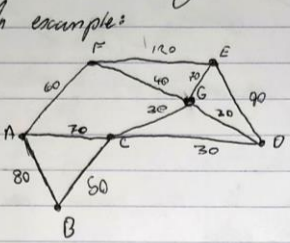
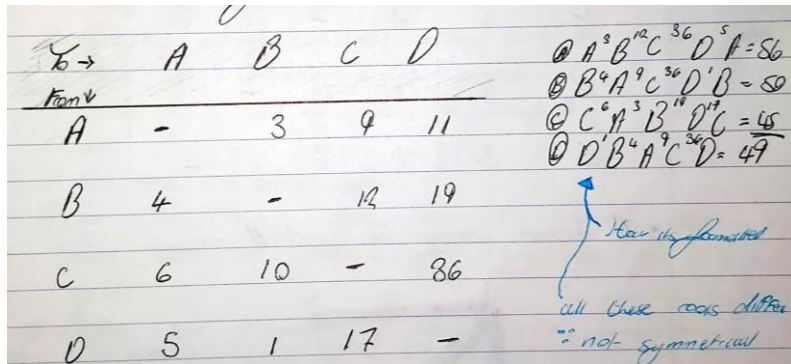


DB - Networks

What are the steps of Prim's Algorithm?	<ol style="list-style-type: none"> 1. Choose a starting vertex. 2. Connect this vertex to the nearest unconnected vertex by the shortest edge. 3. For any already connected vertex, add the shortest edge that connects a new vertex. 4. Repeat step 3 until all the vertices are connected.
What 3 things should you remember when doing Prim's Algorithm on a table?	<ol style="list-style-type: none"> 1. To delete vertices as you as along so you don't form a cycles. 2. Numbering the order of vertices as you go along. 3. Looking down/across all columns/rows after adding one.
What are the steps of Kruskal's Algorithm?	<ol style="list-style-type: none"> 1. Start with the shortest edge. 2. From the remaining edges, select the next shortest edge that does not form a cycle. 3. Repeat step 2 until all the vertices are connected. <p><i>Or this can be done by listing the arcs in ascending order of weight then following the method above.</i></p>
What's the USUAL aim of the Chinese Postman Problem?	To travel along EVERY EDGE (to deliver letters) in the least possible distance and return to the starting vertex.
What does the Chinese Postman Problem usually require?	The graph to be Eulerian.
What are the steps of the Chinese Postman Problem?	<ol style="list-style-type: none"> 1. Write down the odd vertices. 2. Pair the odd vertices and find the shortest distance between each pair (n.b. to focus here!) 3. Find the shortest odd pairing to make the graph Eulerian.

	<p>• An example:</p>  <p>Minimum route starting and ending at A?</p> <p>① Odd = A, F, E, D (There will always be an even no. of odd vertices)</p> <p>② Possible pairs of odd vertices:</p> <ul style="list-style-type: none"> • AF & ED • AE & FD • AD & FE <p>③ Minimum weight of pairings:</p> <ul style="list-style-type: none"> • $AF + ED = 60 + 90 = 150$ • $AE + FD = 170 + 70 = 240$ • $AD + FE = 120 + 110 = 230$ <p>Using as ^{minimum} possible 'roads'</p> <p>Shortest \Rightarrow total weight of optimum Eulerian</p> <p>total: $690 + 150 = 840$</p>
<p>What is the aim of the Travelling Salesperson Problem?</p>	<p>To travel to EVERY VERTEX (to sell items in towns and cities) and return to the starting vertex in the least possible distance.</p> <p><i>This is simply trying to find a Hamiltonian cycle of minimum length.</i></p>
<p>What can you split the Travelling Salesperson Problem into?</p>	<p>Finding a upper bound and a lower bound between which the optimal solution lies.</p>
<p>What are the upper and lower bounds for the Travelling Salesperson Problem?</p>	<ul style="list-style-type: none"> • The upper bound - a total distance for which there DEFINITELY exists a Hamiltonian cycle. • The lower bound - a minimum total distance for which a Hamiltonian cycle MAY exist.
<p>What are the best upper and lower bounds for the Travelling Salesperson Problem and why?</p>	<ul style="list-style-type: none"> • The lowest upper bound because it will definitely exist. • The greatest lower bound because as it cancels all the other lower bounds. <p><i>This is because if a lower bound forms a tour then that tour will be the optimal solution. If none of the possible lower bounds form a tour, the optimal tour must be greater than the best lower bound.</i></p>
<p>What must the graph be in the Travelling Salesperson</p>	<ul style="list-style-type: none"> • Complete.

Problem and how can you make it this?	<ul style="list-style-type: none">By adding indirect edges of minimum length (like in the Chinese Postman Problem).																									
Give a problem with the Travelling Salesperson Heuristic Solution	Doesn't account for if you wanted to visit a vertex twice.																									
How do you find the upper and lower bound of the Travelling Salesperson Problem?	By using the Nearest Neighbour Algorithm and Lower Bound Algorithm respectively.																									
What are the steps for the Nearest Neighbour Algorithm?	<div><div><div>1. Choose a starting vertex on the table.</div><div>2. From this current vertex, go to the nearest unvisited vertex.</div><div>3. Repeat step 2 until all the vertices have been visited.</div><div>4. Return to the starting vertex.</div></div><div><p>The handwritten image shows a distance table for vertices A, B, C, and D. The table is as follows:</p><table><tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th></tr><tr><th>A</th><td>-</td><td>3</td><td>9</td><td>11</td></tr><tr><th>B</th><td>4</td><td>-</td><td>12</td><td>19</td></tr><tr><th>C</th><td>6</td><td>10</td><td>-</td><td>36</td></tr><tr><th>D</th><td>5</td><td>1</td><td>17</td><td>-</td></tr></table><p>Handwritten calculations on the right:</p><ul style="list-style-type: none">$A^3 B^2 C^3 D^5 = 86$$B^4 A^2 C^3 D^1 = 50$$C^6 A^3 B^2 D^1 = 45$$D^5 B^4 A^2 C^3 = 49$<p>Annotations: "Have to find the smallest" with an arrow pointing to the value 1 between B and D. "all these ones differ" and "not symmetrical" with arrows pointing to the non-symmetrical values in the table.</p></div><p><i>This differs from Prim's in that after A, you only look down B and not both.</i></p></div>		A	B	C	D	A	-	3	9	11	B	4	-	12	19	C	6	10	-	36	D	5	1	17	-
	A	B	C	D																						
A	-	3	9	11																						
B	4	-	12	19																						
C	6	10	-	36																						
D	5	1	17	-																						
What are the steps for the Lower Bound Algorithm?	<div><div><div>1. Delete the stated vertex and all the edges connected to it.</div><div>2. Find the minimum spanning tree of the remaining vertices (using Prim's from a table at any start).</div><div>3. Connect the two shortest edges from the deleted vertex to the minimum spanning tree.</div></div><div><p><i>You have to connect two edges from the deleted vertex to try and make it a Hamiltonian cycle.</i></p></div></div>																									
What should you be note for the Nearest Neighbour Algorithm?	<ul style="list-style-type: none">Starting at different vertices may give different no.'s which is why you may have different upper bounds.																									