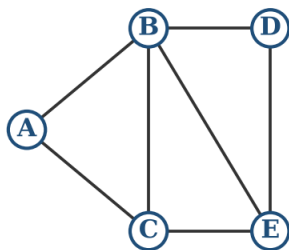


Graph Theory

- 1 Write down the degree of each vertex of the graph shown, and verify that the sum of the degrees is twice the number of edges.



[2]

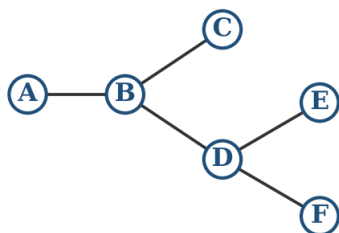
- 2 A simple connected graph has exactly four vertices.

- (a) Write down the least and the greatest number of edges the graph can have. [2]
 (b) Draw a simple connected graph with four vertices and exactly five edges, and state, with a reason, whether it is Eulerian, semi-Eulerian or neither. [2]

- 3 K_6 is the complete graph on 6 vertices.

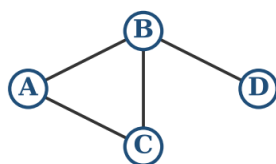
- (a) Write down the degree of each vertex. [1]
 (b) Find the number of edges. [1]

- 4 State whether the graph shown is a tree, giving two reasons.



[2]

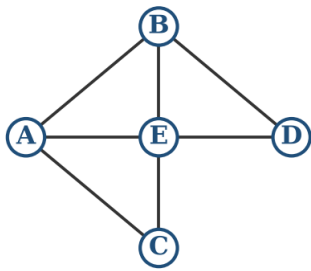
- 5 Write down the adjacency matrix of the graph shown, with the vertices in the order A, B, C, D.



[2]

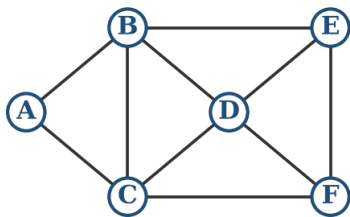
- 6 A connected graph has five vertices with degrees 2, 4, 2, 4, 2. State, with a reason, whether it has an Eulerian cycle. [2]

- 7 For the graph shown, state whether there is a trail that uses every edge exactly once. If there is, give such a trail and state whether it can be closed.



[3]

- 8 The graph shown has 6 vertices. Write down a Hamiltonian cycle for it, and explain what makes the cycle Hamiltonian.



[3]

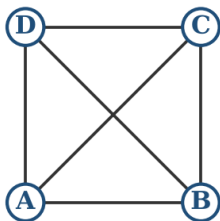
- 9 Prove that it is impossible for a graph to have exactly 5 vertices of odd degree and all other vertices of even degree.

[3]

- 10 $K_{3,4}$ is the complete bipartite graph with vertex sets of sizes 3 and 4, in which every vertex of one set is joined to every vertex of the other.

- (a) Find the number of edges. [2]
 (b) State the degrees of the vertices in each set and determine whether $K_{3,4}$ has an Eulerian trail. [2]

- 11 The complete graph K_4 is drawn with two edges crossing. Show that K_4 is planar by redrawing it so that no edges cross.



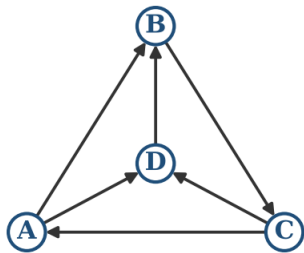
[2]

- 12 A graph has vertices A, B, C, D, E and adjacency matrix (rows and columns in that order)

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

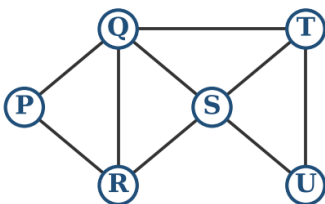
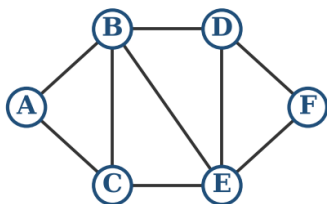
- (a) Find the degree of each vertex and the number of edges. [2]
 (b) List the edges and state whether the graph is connected. [2]

- 13** The directed graph shown has vertices A, B, C and D.



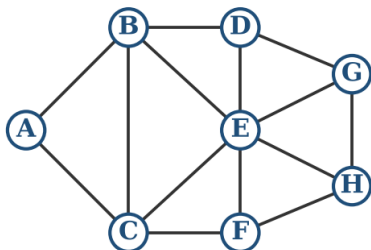
- (a) Find the in-degree and out-degree of each vertex. [2]
 (b) Write down a directed cycle through three vertices. [1]

- 14** The two graphs shown, G_1 with vertices A to F and G_2 with vertices P to U, are isomorphic.



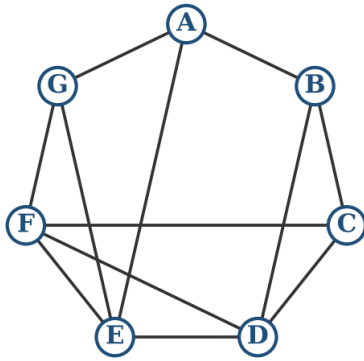
- (a) Show that the two graphs have the same degree sequence. [2]
 (b) Find a mapping from the vertices of G_1 to those of G_2 that preserves adjacency and verify it for three of the edges. [3]

- 15** The planar graph shown is drawn with no edges crossing. Use Euler's formula $V - E + F = 2$ to find the number of faces (including the infinite outer face), and verify by counting the faces in the diagram.



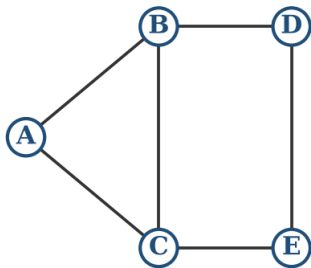
[4]

- 16** The graph shown has the Hamiltonian cycle ABCDEFGA. The other five edges, AE, BD, CF, DF and EG, are called chords.



- (a) Explain why the chords AE and CF cannot both be drawn inside the cycle without crossing, but AE and BD can. [2]
- (b) Use the planarity algorithm, with the cycle ABCDEFGA, to show that the graph is planar, stating which chords are drawn inside and which outside the cycle. [4]

- 17** Show that the graph shown is not bipartite, and explain your method.

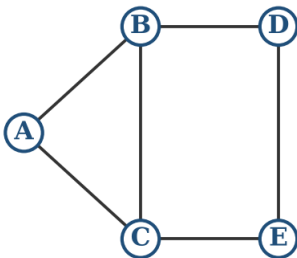


[3]

- 18** A tree T has 8 vertices.

- (a) Find the number of edges and the sum of the degrees of the vertices. [2]
- (b) T has one vertex of degree 4, one of degree 3, one of degree 2 and all the other vertices of degree 1. Find the number of vertices of degree 1 and show that the degrees are consistent with part (a). [3]

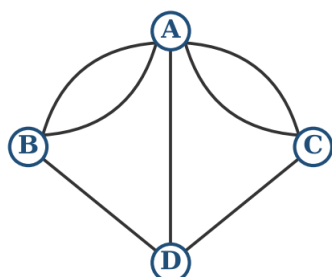
- 19** The graph G shown has 5 vertices. The complement of G has the same vertices, and an edge joins two vertices exactly when they are not joined in G .



- (a) List the edges of the complement of G . [2]
- (b) Write down the degree of each vertex in the complement and hence find the degrees in G . [2]

- 20** In a group of 9 people, each person shakes hands with some of the others, and no two people shake hands more than once.
- (a) Show that it is impossible for each person to shake hands with exactly 3 others. [3]
- (b) Show that it is possible if each person shakes hands with exactly 4 others, by describing a suitable arrangement. [2]

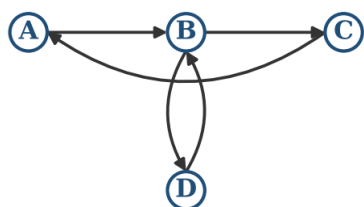
- 21** A town has four districts A, B, C and D, joined by seven bridges as shown. Each bridge is an edge of a graph and each district is a vertex.



- (a) Show that it is impossible to start in one district and cross every bridge exactly once, whatever the start and finish. [3]
- (b) One new bridge is built, joining two of the districts, so that a walk crossing every bridge exactly once becomes possible. Find all the possible pairs of districts, and state where the walk would start and finish in each case. [3]
- 22** For a connected planar bipartite graph with $V \geq 3$ vertices and E edges, every face is bounded by at least 4 edges, and it can be shown that $E \leq 2V - 4$.
- (a) Show that $K_{3,3}$ is not planar. [3]
- (b) Show that the inequality is satisfied by $K_{2,4}$, and draw $K_{2,4}$ to show it is planar. [3]
- (c) Hence explain why satisfying the inequality does not by itself prove that a graph is planar. [1]
- 23** A simple graph has 6 vertices.
- (a) Show that the degree sequence 4, 4, 3, 3, 2, 2 is possible by drawing a graph with these degrees, and state its number of edges. [3]
- (b) Show that no simple graph on 6 vertices has degrees 5, 5, 4, 3, 2, 1. [3]

- 24** A tree is a connected graph with no cycles.
- (a) Explain why every tree with at least 2 vertices has a vertex of degree 1. [2]
- (b) Hence prove by induction that a tree with n vertices has $n - 1$ edges. [4]

- 25** The directed graph shown represents one-way roads between four junctions.



- (a) Show that every junction has the same number of roads leaving as entering. [2]
- (b) Find a route that starts and ends at A and uses every road exactly once. [2]

- 26** A graph has vertices A, B, C, D and adjacency matrix

$$\begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

The entry in row i and column j of the matrix A^n is the number of walks of length n from vertex i to vertex j .

- Find A^2 . [2]
- Write down the number of walks of length 2 from A to D, and list them. [2]
- Find the number of walks of length 3 from A to A. [2]

- 27** $K_{m,n}$ is the complete bipartite graph with vertex sets of sizes m and n .

- Show that $K_{3,4}$ has no Hamiltonian cycle. [2]
- Show that $K_{3,3}$ has exactly 6 distinct Hamiltonian cycles (a cycle and its reverse count as the same). [3]
- Show that $K_{4,4}$ has 72 distinct Hamiltonian cycles. [3]

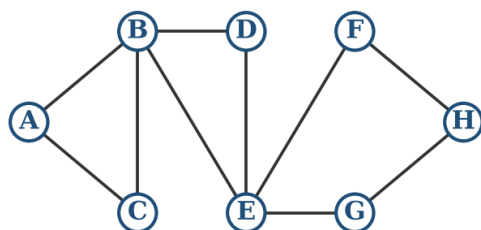
- 28** A connected simple planar graph has V vertices, E edges and F faces, so that $V - E + F = 2$. Each face is bounded by at least 3 edges and each edge borders at most 2 faces. Assume $V \geq 3$.

- Show that $3F \leq 2E$ and hence that $E \leq 3V - 6$. [4]
- Hence show that K_5 is not planar. [2]
- Show that this inequality does not prove that $K_{3,3}$ is non-planar, and state a stronger inequality that does if the graph is bipartite. [2]

- 29** A student makes three claims about connected simple graphs. For each claim, state whether it is true or false and justify your answer, using a counterexample where it is false.

- Claim 1: "If every vertex has even degree, the graph has an Eulerian cycle." [2]
- Claim 2: "If a graph has a Hamiltonian cycle then every vertex has degree at least 2." [1]
- Claim 3: "If every vertex has degree at least 2 then the graph has a Hamiltonian cycle." [2]
- Claim 4: "A tree with 10 vertices can have a vertex of degree 9 and also a vertex of degree 8." [2]

- 30** The graph shown has 8 vertices and 10 edges.

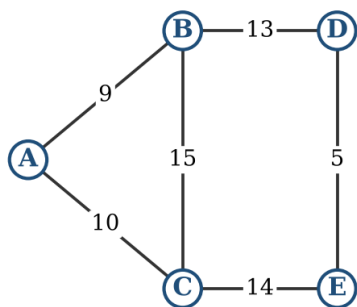


- Show that the graph has an Eulerian cycle. [2]
- Find an Eulerian cycle starting and ending at A. [3]
- Explain why a graph in which all degrees are even can always be split into cycles, and hence why every connected such graph has an Eulerian cycle. [3]

Minimum Spanning Trees

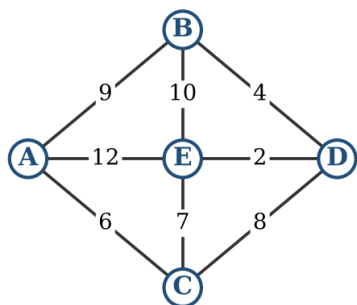
- 1** State what is meant by a spanning tree of a connected network. A connected network has 8 vertices: how many edges does a spanning tree of it contain? [2]

- 2 Use Kruskal's algorithm to find a minimum spanning tree for the network shown. State the edges chosen in order and the total weight.



[3]

- 3 Use Prim's algorithm, starting at A, to find a minimum spanning tree for the network shown. State the order in which the vertices are added and the total weight.



[3]

- 4 Kruskal's algorithm is being used on a network with vertices A, B, C, D, E, F. The edges AB, CD and BE have been accepted so far. For each of the edges below, state whether Kruskal's algorithm would accept it next, and give a reason.

- (a) AE
(b) DE
(c) CF

[1]

[1]

[1]

- 5 The table shows the distances in kilometres between four villages. Find the minimum total length of cable needed to connect all four villages.

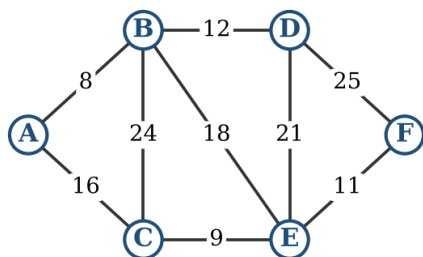
	A	B	C	D
A	-	7	4	9
B	7	-	5	8
C	4	5	-	6
D	9	8	6	-

[3]

- 6 The weights of the edges of a network on vertices A to E are listed.
AB 12, AC 8, AD 15, BC 6, BE 10, CD 9, CE 14, DE 7
Write the edges in the order they are considered by Kruskal's algorithm and state which are rejected.

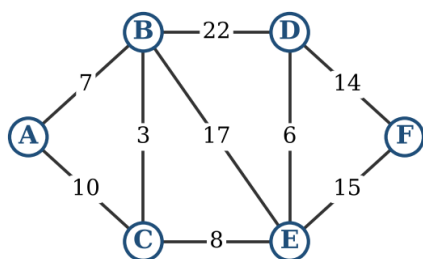
[2]

- 7 Use Kruskal's algorithm to find a minimum spanning tree for the network shown. List the edges in the order in which you consider them, state which you reject and give the total weight of the tree.



[4]

- 8 Use Prim's algorithm, starting at A, to find a minimum spanning tree for the network shown. State the order in which the vertices are added, the edges chosen and the total weight.



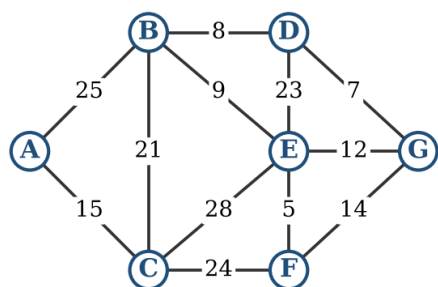
[4]

- 9 The table shows the direct distances, in kilometres, between six towns A to F (a dash means no direct road). Use Prim's algorithm, starting at A, to find a minimum spanning tree, and state its total length.

	A	B	C	D	E	F
A	-	13	6	-	-	-
B	13	-	4	24	16	-
C	6	4	-	17	-	12
D	-	24	17	-	9	23
E	-	16	-	9	-	15
F	-	-	12	23	15	-

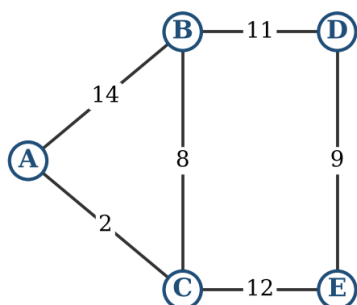
[4]

- 10 Use Kruskal's algorithm to find a minimum spanning tree for the network shown, and state its total weight.



[4]

- 11** Six villages are to be connected by a cable network. The possible links and their lengths in kilometres are: AB 5.2, AC 3.8, BC 4.1, BD 6.5, CD 3.4, CE 7.0, DE 4.6, DF 5.9, EF 3.9. The cable costs £2400 per kilometre.
- (a) Find the minimum total length of cable needed to connect all six villages, and the links used. [3]
- (b) Find the minimum cost of the cable. [1]
-
- 12** A network consists of five vertices joined in a cycle ABCDEA, with edge weights AB = 6, BC = 4, CD = 7, DE = 5, EA = 8.
- (a) Show that the network has exactly 5 spanning trees. [2]
- (b) Hence find the minimum spanning tree and its weight. [2]
-
- 13** The network shown shows the possible capacities of five pipes. Find a maximum spanning tree, that is a spanning tree of greatest total weight, using a version of Kruskal's algorithm, and state its weight.



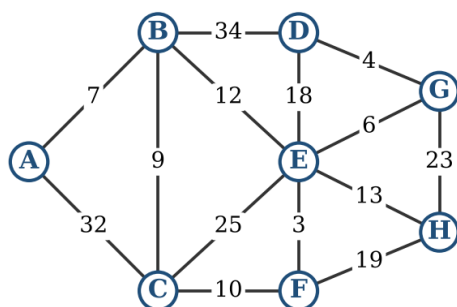
[3]

- 14** The table shows the distances between seven sites. Use Prim's algorithm, starting at D, to find a minimum spanning tree, giving the order in which vertices are added, and draw the tree (describe it by listing the edges). State the total weight.

	A	B	C	D	E	F	G
A	-	4	19	-	-	-	-
B	4	-	7	28	16	-	-
C	19	7	-	-	8	25	-
D	-	28	-	-	5	-	30
E	-	16	8	5	-	27	21
F	-	-	25	-	27	-	11
G	-	-	-	30	21	11	-

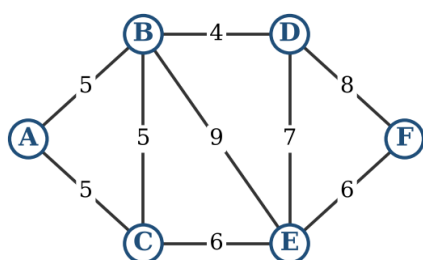
[5]

- 15** Use Kruskal's algorithm to find a minimum spanning tree for the network shown. State the order in which edges are accepted, the edges rejected and the total weight of the tree.



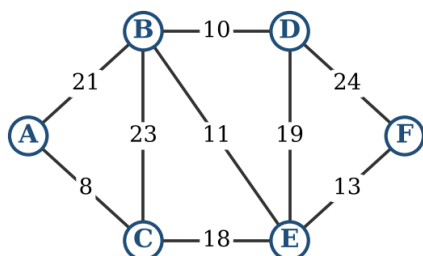
[4]

- 16** The network shown has several edges of equal weight.



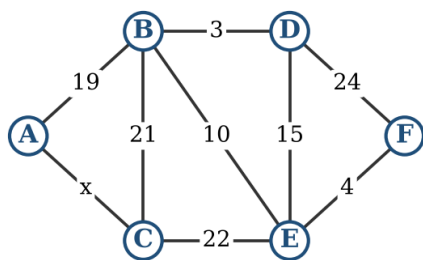
- (a) Use Kruskal's algorithm to find a minimum spanning tree and its weight. [4]
 (b) Show that the minimum spanning tree is not unique, by finding a second one of the same weight. [2]

- 17** The network shown is to be connected by a minimum spanning tree.



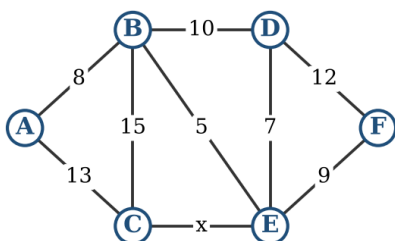
- (a) Find the minimum weight of a spanning tree, using Kruskal's algorithm. [4]
 (b) For planning reasons the edge DF (weight 24) must be included. Find the minimum weight of a spanning tree that includes it. [3]

- 18** In the network shown, the weight of edge AC is x , where x is a positive integer. When $x = 14$ the edge is part of a minimum spanning tree.



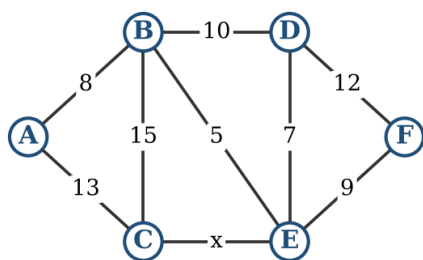
- (a) Use Kruskal's algorithm on the other edges to find a minimum spanning tree of the network without edge AC, and the path in it between A and C. [3]
 (b) Hence find the greatest value of x for which AC can be in a minimum spanning tree. [2]

- 19** The network shown connects six towns. A seventh town G is to be added, with possible links GB (9), GD (7) and GA (12).



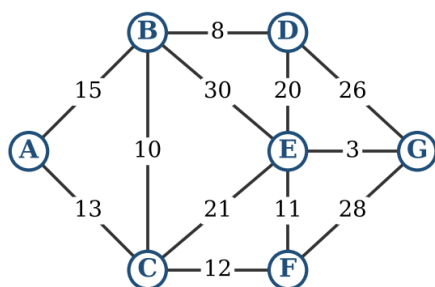
- (a) Find the weight of a minimum spanning tree of the original network of six towns. [4]
 (b) Find the weight of a minimum spanning tree of the new network and explain how it is related to your answer to (a). [4]

- 20** The network shown has the weight of arc CE equal to x , where x is a positive integer. Prim's algorithm, starting at A, adds the arcs to the minimum spanning tree in the order AB, BE, ED, EF, EC.



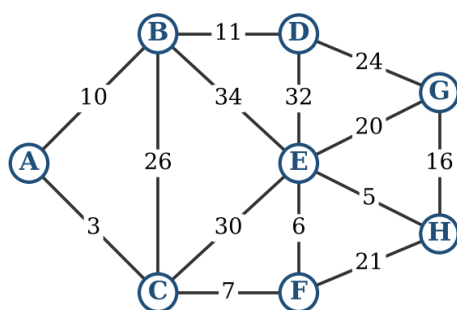
- (a) Use the order in which the arcs are added to show that $9 < x < 13$. [3]
 (b) The minimum spanning tree has total weight 40. Find the value of x . [2]

- 21** Both Kruskal's and Prim's algorithms are applied to the network shown.



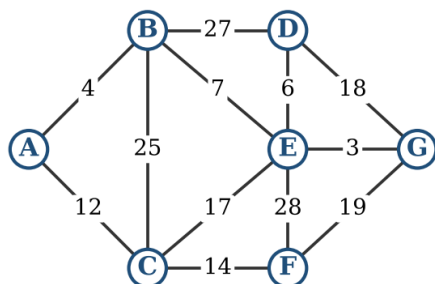
- Apply Kruskal's algorithm and state the minimum total weight. [4]
- Apply Prim's algorithm starting at A and state the order in which vertices are added. [3]
- Explain why the two algorithms give trees of the same weight, and state one situation in which Prim's algorithm is more convenient than Kruskal's. [2]

- 22** The network shown has a minimum spanning tree T.



- Use Kruskal's algorithm to find T and its weight. [4]
- The road CF (weight 7) is closed. Find the new minimum spanning tree and its weight, and show how it is obtained from T by replacing one edge. [4]

- 23** The network shown has a unique minimum spanning tree.



- Find the minimum spanning tree and its weight. [4]
- The weight of the edge CF (currently 14) is increased to y . Find the greatest value of y for which CF is still in a minimum spanning tree. [3]
- Find the minimum spanning tree weight when $y = 24$. [2]

- 24** Prove the following statements about a connected network with distinct edge weights.

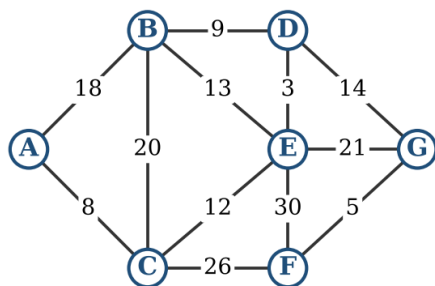
- Kruskal's algorithm never creates a cycle. [2]
- The minimum spanning tree is unique. [3]

- 25** The table shows the cost in thousands of pounds of laying a direct power line between eight villages A to H; a dash means that no direct line is possible.

	A	B	C	D	E	F	G	H
A	-	19	14	-	-	-	-	-
B	19	-	11	30	34	-	-	-
C	14	11	-	-	33	21	-	-
D	-	30	-	-	12	-	31	-
E	-	34	33	12	-	16	18	27
F	-	-	21	-	16	-	-	20
G	-	-	-	31	18	-	-	35
H	-	-	-	-	27	20	35	-

- (a) Use Prim's algorithm, starting at A, to find the cheapest way of connecting all eight villages, giving the order in which villages are added and the lines laid. [4]
- (b) A new line HC, costing 9, is now possible. Find the new minimum cost of connecting all eight villages. [3]

- 26** The network shown has minimum spanning tree T. A new road AG of length 12 is proposed.

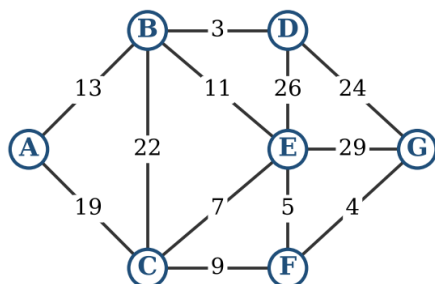


- (a) Find the weight of T and the path in T from A to G. [4]
- (b) Explain, using the path in T, whether the new road changes the minimum spanning tree, and find the new minimum weight. [3]

- 27** Cayley's formula states that the number of spanning trees of the complete graph K_n (n labelled vertices, every pair joined) is n^{n-2} .

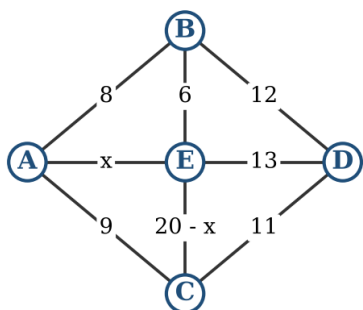
- (a) Verify the formula for $n = 4$ by counting the spanning trees of K_4 directly. [4]
- (b) Find the number of spanning trees of K_6 , and the number that are stars (one vertex joined to all the others). [2]
- (c) Find the number of spanning trees of $K_{2,3}$ (vertex sets of sizes 2 and 3, every pair across the sets joined) and show that it is 12. [4]

- 28** The network shown has minimum spanning tree T . The bottleneck of a route is the weight of its heaviest edge.



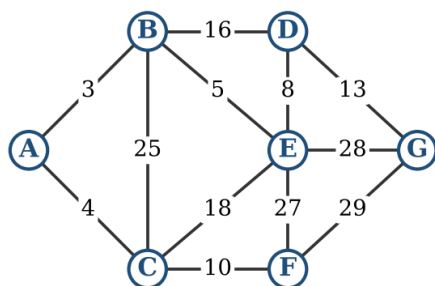
- (a) Find T and the route in T from A to G . State its bottleneck. [4]
 (b) Show that any route from A to G in the network has bottleneck at least 13. [3]
 (c) State the general property of a minimum spanning tree that this illustrates. [1]

- 29** In a network on five vertices, $AB = 8$, $AC = 9$, $BE = 6$, $BD = 12$, $CD = 11$, $DE = 13$, $AE = x$ and $CE = 20 - x$, where x is an integer with $1 \leq x \leq 19$. The layout is as in the diagram.



- (a) Find the weight of the minimum spanning tree when $x = 5$ and when $x = 15$. [3]
 (b) Find the value of x for which the minimum spanning tree has its least possible weight, and state that weight. [3]

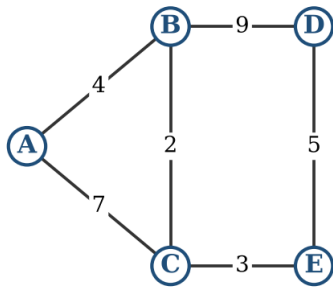
- 30** The network shown is to be connected by a spanning tree. A planning rule says that vertex A must be an end of the tree, that is A must have exactly one edge of the tree attached.



- (a) Find the weight of the minimum spanning tree with no extra rule, and state how many edges at A it uses. [5]
 (b) Show that the cheapest tree satisfying the rule has weight 57, and explain the method. [5]

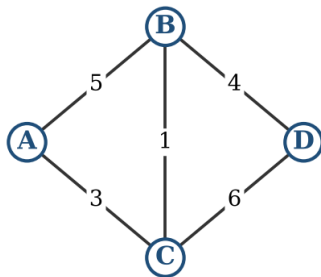
Shortest Path Algorithms

- 1 Find the shortest route from A to E in the network shown, and state its length.



[2]

- 2 List all the routes from A to D that do not visit any vertex twice, with their lengths, and state the shortest.



[3]

- 3 In Dijkstra's algorithm, state what the final label of a vertex represents.
In the network above, after vertex A is given the final label 0 the working values are B 5 and C 3. State which vertex receives the next final label, and its value.

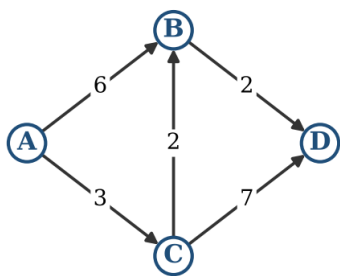
[2]

- 4 The table shows the direct distances between four towns. Find the shortest route from A to D and its length.

	A	B	C	D
A	-	6	2	9
B	6	-	3	5
C	2	3	-	8
D	9	5	8	-

[2]

- 5 The directed network shows one-way roads and their lengths. Find the shortest route from A to D and its length.

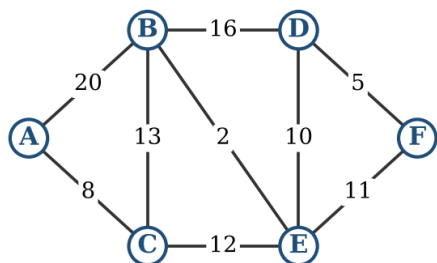


[2]

- 6 Dijkstra's algorithm is used on a network. The working values at one stage are: B 12, C 9, D 15, E 11, and these vertices have not yet been given final labels. State which vertex is labelled next.

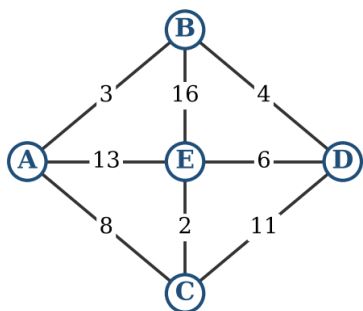
[2]

- 7 Use Dijkstra's algorithm to find the shortest route from A to F in the network shown. Show your working values and give the length of the route.



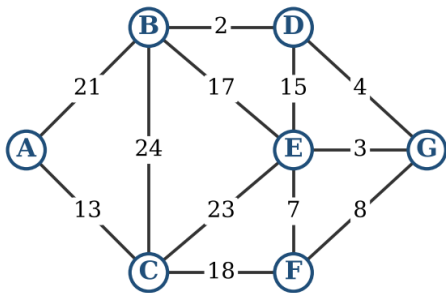
[5]

- 8 Use Dijkstra's algorithm to find the shortest route from A to D in the network shown, and state its length.



[4]

- 9 Use Dijkstra's algorithm to find the shortest route from A to G in the network shown. Show the order of labelling, the final labels and the working values, and state the length of the route.



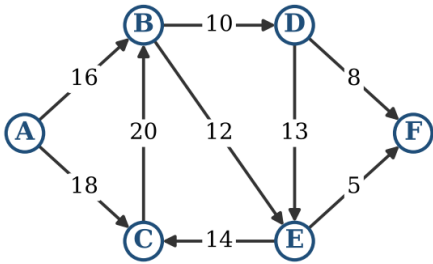
[5]

- 10 The table shows the lengths, in kilometres, of direct roads between six towns (a dash means that there is no direct road). Use Dijkstra's algorithm to find the shortest route from A to F and state its length.

	A	B	C	D	E	F
A	-	5	15	-	-	-
B	5	-	3	19	13	-
C	15	3	-	22	-	12
D	-	19	22	-	4	8
E	-	13	-	4	-	18
F	-	-	12	8	18	-

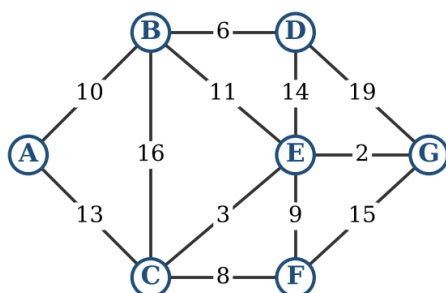
[4]

- 11 The directed network shows one-way streets and the time in minutes to travel along each. Use Dijkstra's algorithm to find the quickest route from A to F, and state its time.



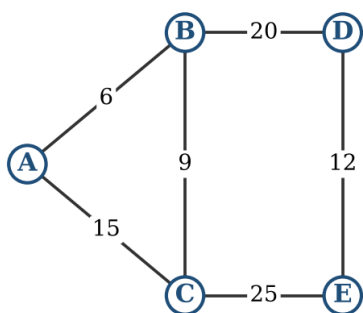
[4]

- 12** A delivery driver must go from A to G, calling at E on the way. The network shows the distances in kilometres.



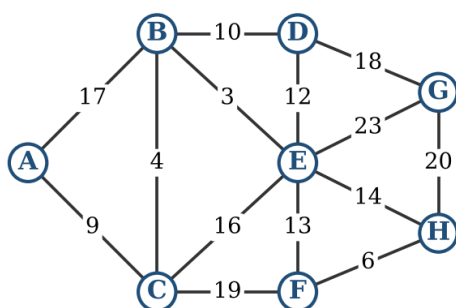
- (a) Use Dijkstra's algorithm to find the shortest distance from A to E and the route. [2]
 (b) Find the shortest route from E to G and hence the shortest route from A to G that calls at E. [3]

- 13** The network shows the lengths in kilometres of roads between five places. A cyclist averages 20 km per hour.



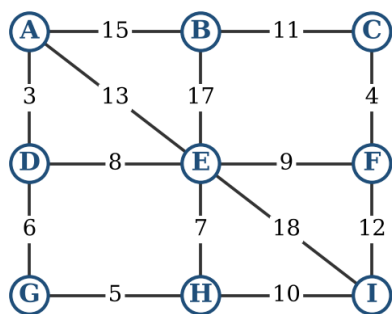
- (a) Find the shortest route from A to E and its length. [4]
 (b) Find the time for the journey, in minutes. [2]

- 14** Use Dijkstra's algorithm to find the shortest route from A to H in the network shown. Show the order of labelling, final labels and working values, and state the length of the route.



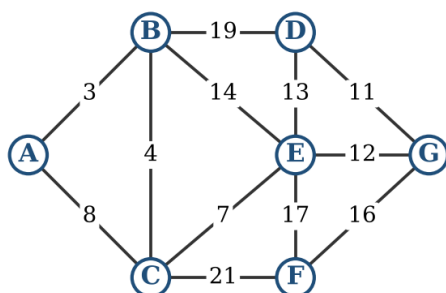
[6]

- 15** Use Dijkstra's algorithm to find the shortest route from A to I in the network shown, giving the length of the route. Show the order of labelling, the final labels and all working values.



[6]

- 16** The network shows distances between seven depots.

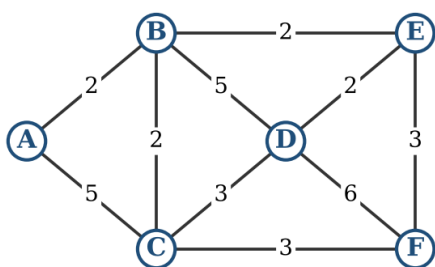


- Use Dijkstra's algorithm to find the shortest route from A to G and its length.
- Depot C is closed and cannot be used. Find the shortest route from A to G that avoids C.

[5]

[3]

- 17** In the network shown there are several shortest routes from A to F.

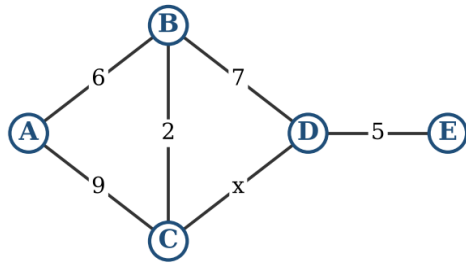


- Use Dijkstra's algorithm to find the length of the shortest route from A to F.
- Find all the routes of this length from A to F.

[4]

[2]

- 18** In the network shown, the length of edge CD is x , where $x > 0$.



- (a) Show that when $x = 3$ the shortest route from A to E has length 16. [2]
 (b) Find the range of values of x for which A-B-C-D-E is the only shortest route from A to E. [2]
 (c) Find the length of the shortest route from A to E when $x = 10$. [1]

- 19** One application of Dijkstra's algorithm to a network with n vertices is of order n^2 . A computer produces a table of the shortest distances between every pair of vertices by applying Dijkstra's algorithm in turn with each vertex as the start. It takes 0.15 seconds to produce the table for a network with 60 vertices.

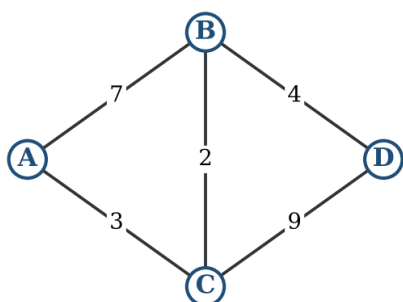
- (a) Show that the whole process is of order n^3 . [2]
 (b) Estimate the time the computer will need for a network with 240 vertices. [2]
 (c) Explain why your answer to part (b) can only be an approximation. [1]

- 20** Floyd's algorithm has been applied to a road network on five towns A to E, with all roads two-way. The final distance table and route table are shown. In the route table, the entry in row X and column Y is the vertex that follows X on a shortest route from X to Y.

Distance	A	B	C	D	E	Route	A	B	C	D	E
A	-	10	7	14	11	A	-	B	C	C	C
B	10	-	6	13	10	B	A	-	C	C	C
C	7	6	-	7	4	C	A	B	-	E	E
D	14	13	7	-	3	D	E	E	E	-	E
E	11	10	4	3	-	E	C	C	C	D	-

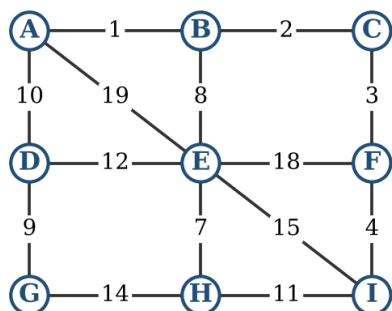
- (a) Explain how the route table is used to find the shortest route from D to A, and state the route and its length. [3]
 (b) Use the nearest neighbour algorithm on the distance table, starting at A, to find an upper bound for the length of a tour that visits every vertex once and returns to A. [3]

- 21** Floyd's algorithm is used to find the shortest distance between every pair of vertices of the network shown. At each iteration k , for every pair (i, j) , the distance $d(i, j)$ is replaced by $d(i, k) + d(k, j)$ if this is smaller.



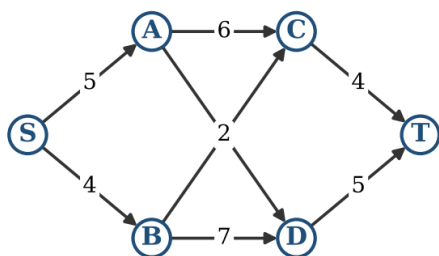
- Write down the initial distance matrix, with ∞ for pairs of vertices that are not joined directly. [2]
- Complete the four iterations, showing the matrix after each, and write down the final matrix. [4]
- Hence write down the shortest distance from A to D, and the route. [2]

- 22** The network shown represents the times in minutes between nine junctions.



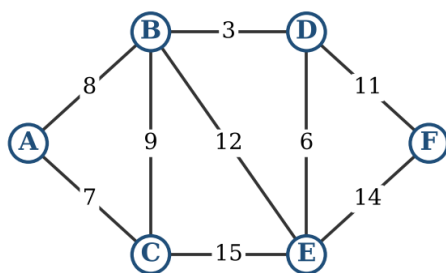
- Use Dijkstra's algorithm to find the quickest route from A to I, and the time taken. [6]
- Find the vertices which are further from A than I by the quickest route, if any, and state the vertex furthest from A. [2]

- 23** The directed network shows the times in hours for tasks that must be done in the direction of the arrows, from S to T.



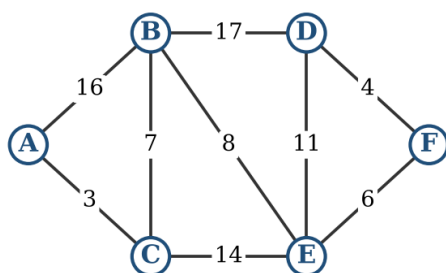
- (a) Dijkstra's algorithm can be adapted to find the longest route by choosing the largest working value. Explain why this adaptation works for this network, but would not work for a general network with cycles. [2]
- (b) Find the longest route from S to T and its length, showing the largest label at each vertex. [4]

- 24** The network shows journey times in minutes between junctions. A vehicle also loses time waiting at each intermediate junction (not at the start or finish): B 2, C 4, D 1, E 3 minutes.



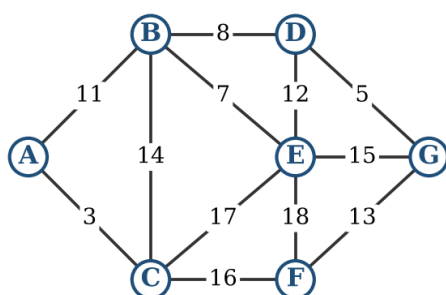
- (a) Use Dijkstra's algorithm to find the quickest route from A to F ignoring the waiting times, and its time. [4]
- (b) Find the quickest route from A to F when the waiting times are included, by adding the waiting time at a junction to every edge leaving it and running Dijkstra's algorithm again (or otherwise), and compare with (a). [4]

- 25** A student applies Dijkstra's algorithm to the network shown, starting at A. The student labels the vertices in this order (final labels in brackets): A (0), B (10), C (3), E (17), F (23), D (27). The student's list contains an error.



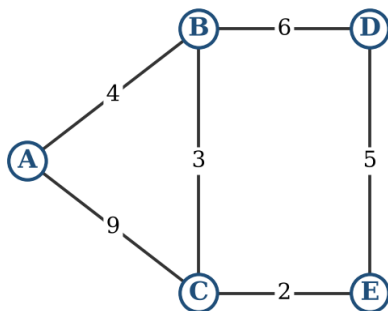
- (a) Identify the error in the order of labelling and explain what rule has been broken. [2]
- (b) Complete the algorithm correctly and find the shortest route from A to F. [4]

- 26** The network shows distances in kilometres. A courier must travel from A to G and must call at both C and E, in either order.



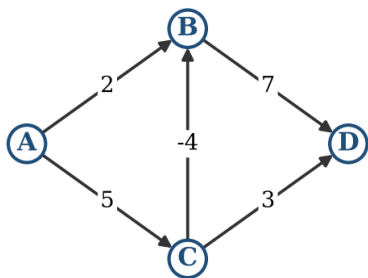
- (a) Use Dijkstra's algorithm from A to find the shortest distances from A to C and from A to E. [2]
- (b) Find the shortest distance from C to E, from C to G and from E to G (using Dijkstra's algorithm or otherwise). [3]
- (c) Hence find the shortest distance for the courier's journey, and the order in which C and E are visited. [3]

- 27 Floyd's algorithm is used on the network shown with vertices A to E (all roads two-way).



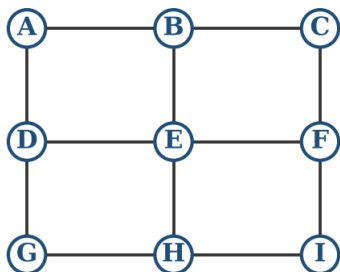
- Write down the initial distance matrix (∞ for no direct link). [2]
- Show the distance matrix after the iteration for vertex B, and after the iteration for vertex C. [4]
- Complete the algorithm. Write down the final matrix, and the shortest distance and route from A to E. Check your distance using Dijkstra's algorithm. [4]

- 28 The directed network shown has one edge of negative weight, $CB = -4$. Dijkstra's algorithm is applied from A as if all weights were non-negative.



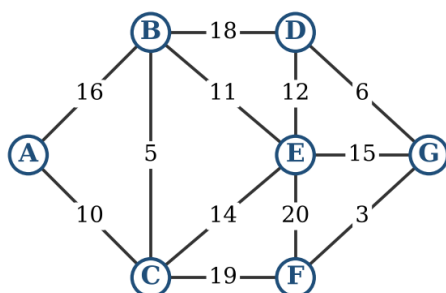
- Apply Dijkstra's algorithm and state the final labels it gives for B and D. [3]
- Show that the algorithm's answer for B is wrong by finding a shorter route to B, and give the true shortest distances to B and D. [3]
- Explain why the argument that justifies Dijkstra's algorithm fails when weights can be negative. [2]

- 29 The network shown is a 3 by 3 grid of streets; each street section between neighbouring junctions has length 1.



- State the length of the shortest route from A to I and explain why every such route has this length. [2]
- Label each junction with the number of shortest routes from A to it, building up from A, and hence find the number of shortest routes from A to I. [4]

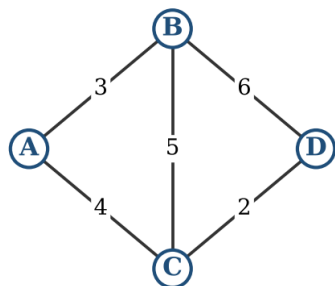
- 30** The network shown gives distances in kilometres. The council will build one new road, of length 6 km, joining two junctions, neither of which is A or G, that are not currently joined directly.



- (a) Use Dijkstra's algorithm to find the shortest distance from A to G at present, and the route. [5]
- (b) Find the new road that gives the shortest A to G distance, justifying your choice, and find that distance. Consider the distances from A and to G of the junctions the road could join. [4]

Route Inspection and Network Problems

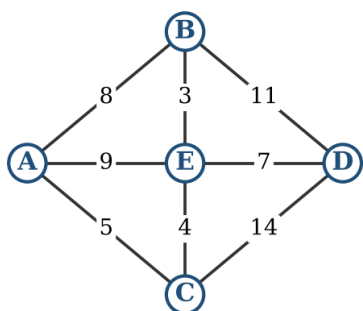
- 1** The network shown has two vertices of odd degree. State which they are and explain why a route that traverses every edge at least once and starts and ends at the same vertex must repeat at least one edge.



[2]

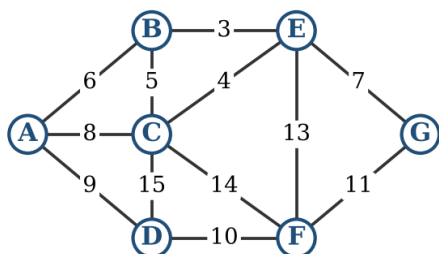
- 2** A network has vertices of degrees 4, 3, 3, 2, 2, 2. State whether it is Eulerian, semi-Eulerian or neither, and how many odd vertices there are. [2]
- 3** The total weight of all edges of a network is 86 and the shortest repeated length needed in a route inspection is 14. Find the length of the shortest closed route that traverses every edge. [2]
- 4** A network has four odd vertices P, Q, R, S. List all the ways of pairing them. [2]
- 5** A salesperson must visit each of 5 towns once, returning to the start. All towns are linked directly. Find the number of different tours (a tour and its reverse count as the same). [2]
- 6** State the difference between the classical travelling salesperson problem and the practical travelling salesperson problem. [2]

- 7 Find the length of a shortest closed route that traverses every edge of the network shown at least once, starting and ending at A. State which edges are repeated.



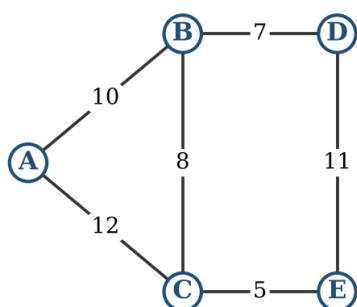
[4]

- 8 Use the route inspection (Chinese postman) algorithm to find the length of the shortest route that traverses every road in the network at least once, starting and finishing at A.



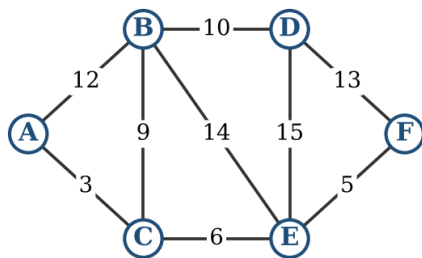
[4]

- 9 The network shows the lengths in metres of paths in a park. A litter collector must walk along every path at least once, starting and finishing at the same place. Find the length of the shortest such walk.



[4]

- 10** The network shown has two odd vertices.



- (a) Write down the odd vertices and the shortest route between them. [2]
 (b) Hence find the length of the shortest closed route traversing every edge, and the shortest route if the start and finish may be at different vertices. [2]

- 11** The table shows the distances between four towns. Use the nearest neighbour algorithm, starting at A, to find a tour of all four towns, returning to A, and state its length.

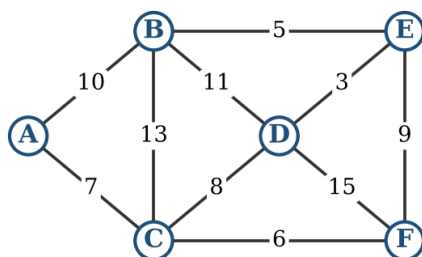
	A	B	C	D
A	-	12	9	15
B	12	-	7	10
C	9	7	-	8
D	15	10	8	-

[3]

- 12** A tour of 5 towns has length 62 km. A different tour, found by a student, has length 58 km, and the student claims that 58 km is an upper bound for the optimal tour.

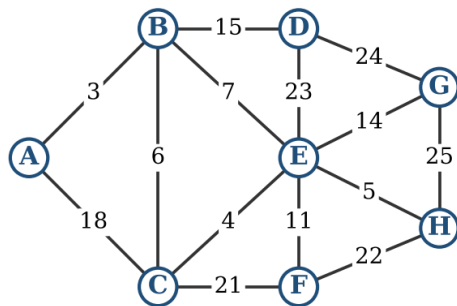
- (a) Explain why a length of any valid tour gives an upper bound for the optimal tour. [1]
 (b) State the best upper bound from the two tours and explain what it tells you about the optimal tour length. [2]

- 13** The network shown has exactly two odd vertices, and the total weight of all edges is 87.



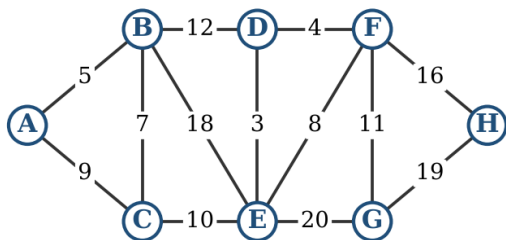
- (a) Find the shortest distance between the two odd vertices E and F. [2]
 (b) Hence find the length of the shortest closed route traversing every edge. [2]

- 14** Find the length of the shortest closed route that traverses every edge of the network shown at least once, starting and ending at A. You must show the pairings you consider and the shortest distance for each.



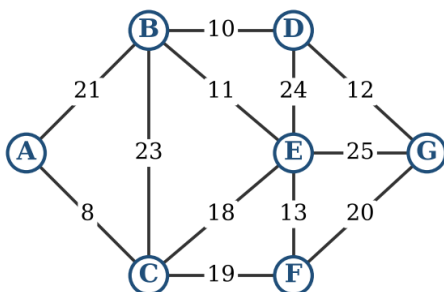
[4]

- 15** Find the length of the shortest closed route that traverses every edge of the network shown at least once. Show the pairings you consider.



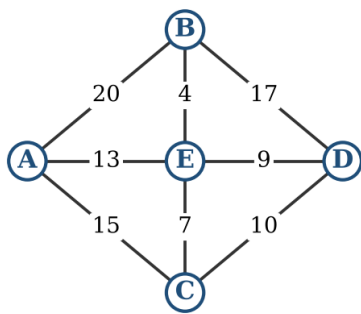
[4]

- 16** The network shown represents the roads in a village.



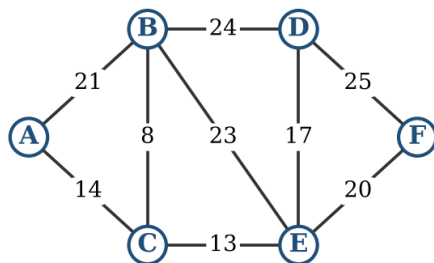
- (a) Find the length of the shortest route that traverses every road at least once, starting and finishing at the same vertex. [4]
- (b) The route may now start at one vertex and finish at another. Find the shortest such route, stating the start and finish. [3]

- 17** The network shown gives the direct distances between five towns. The classical travelling salesperson problem is to be solved, using shortest distances between towns where no direct road exists.



- (a) Complete a table of shortest distances between all pairs of towns. [2]
 (b) Use the nearest neighbour algorithm starting at A to find an upper bound for the optimal tour. [2]

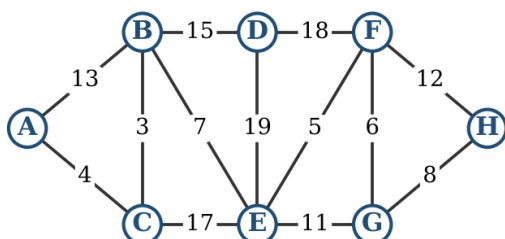
- 18** The table of shortest distances for six towns is obtained from the network shown.



- (a) Use the nearest neighbour algorithm, starting at A, to find an upper bound for the length of the optimal tour. [3]
 (b) Delete vertex E and find a minimum spanning tree of the remaining towns (using shortest distances). Add the two shortest edges at E to obtain a lower bound. [4]
 (c) State the range within which the optimal tour length lies. [1]

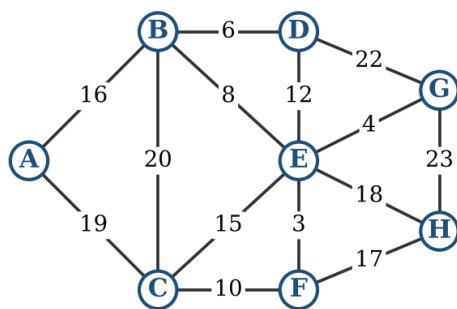
- 19** Explain why the lower bound found by deleting a vertex, finding the minimum spanning tree of the rest and adding the two shortest edges at the deleted vertex is a valid lower bound for the optimal tour. [3]

- 20** A snow plough must clear every road in the network shown, starting and finishing at the depot A. Find the shortest distance it must travel and, given it travels at an average of 30 km per hour, the time in minutes (distances are in km).



[5]

- 24** The network shown has four odd vertices. A route that traverses every edge at least once is to be found.



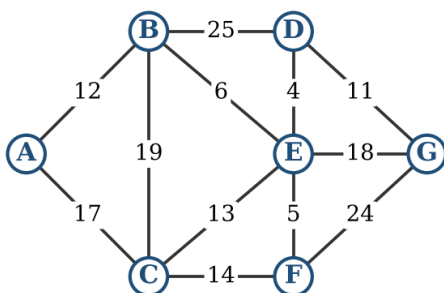
- Find the length of the shortest route that traverses every edge, starting and ending at the same vertex. [4]
- Find the length of the shortest route if the start and finish may be different vertices. [3]
- Hence find how much shorter the open route is than the closed route. [1]

- 25** The table shows the distances between six towns.

	A	B	C	D	E	F
A	-	22	20	23	22	23
B	22	-	12	24	15	24
C	20	12	-	18	22	22
D	23	24	18	-	17	24
E	22	15	22	17	-	13
F	23	24	22	24	13	-

- Apply the nearest neighbour algorithm starting at each of A, B and C, and state the three tour lengths. [4]
- Find a minimum spanning tree and use it to write down a lower bound for the length of any tour. Explain why it is a lower bound. [3]

- 26** The network shown connects seven towns. Distances in km. A tour that visits every town once is required (using shortest distances).



- Find an upper bound by the nearest neighbour algorithm starting at A. [3]
- Find a lower bound by deleting vertex A. [4]
- Deduce the greatest possible percentage error if the nearest neighbour tour is used instead of the optimal tour, to 3 significant figures. [2]

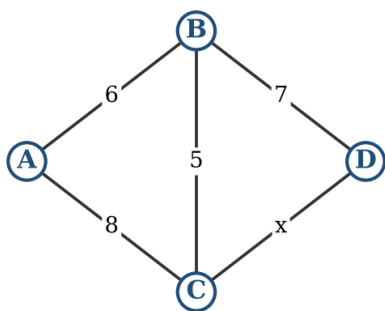
- 27** A connected network has n vertices, all of even degree, and total edge weight W .
- Explain why the shortest closed route traversing every edge has length W . [2]
 - A network has exactly two odd vertices X and Y . Show that the shortest closed route has length $W + d(X, Y)$, where d is the shortest distance between X and Y , and explain why this is the least possible. [4]
 - Explain why a network cannot have exactly three odd vertices. [1]

- 28** The table shows the distances between five towns.

	A	B	C	D	E
A	-	15	14	8	12
B	15	-	16	10	7
C	14	16	-	19	20
D	8	10	19	-	11
E	12	7	20	11	-

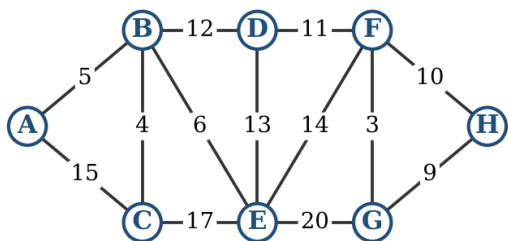
- Use the nearest neighbour algorithm from A to find a tour and its length. [3]
- Find a tour shorter than that in (a) and its length, and use it to show that the nearest neighbour algorithm does not always find the optimal tour. [3]
- Show by checking all $(5 - 1)!/2 = 12$ tours, or otherwise, that 56 is the minimum. [2]

- 29** In the network shown, the length of edge CD is x km, where x is a positive integer. The odd vertices are B and C.



- Show that the shortest distance between B and C is 5 for every positive integer x . [2]
- Hence show that the length of the shortest closed route traversing every edge is $31 + x$. [3]
- Given that the shortest route is 40 km, find x , and state the repeated edge. [3]

- 30** The network shown has four odd vertices and total weight 139.



- Find the shortest closed route traversing every edge. [4]
- List the totals of the three possible pairings and find the percentage by which the longest closed route obtained by the algorithm's pairings exceeds the shortest, to 3 significant figures. [4]
- Explain how the answer would change if one of the edges repeated in your solution were permanently closed (state the general approach only). [2]