IB Math AI HL — Topic 3

Graph Theory

Intro to Graphs · Walks & Adjacency Matrices · Kruskal's · Prim's · Chinese Postman · Travelling Salesman · Nearest Neighbour & Deleted Vertex

Difficulty	EASY	MEDIUM	HARD
Questions	10	10	10
Marks each	3	4	5
Total Marks	30	40	50

Scan me for help

Name:____________________________ Date:______________

Score:_________/120

Instructions: Answer all questions. Show all working clearly. Give answers correct to 3 significant figures unless stated otherwise. GDC permitted throughout. Diagrams are *not to scale* unless stated.

Key Formulae — Graph Theory

1. **Degree of a vertex** = number of edges incident to that vertex. Sum of all degrees $= 2 \times$ (number of edges).
2. **Eulerian circuit**: exists $\Leftrightarrow$ graph is connected & *every* vertex has even degree. **Semi-Eulerian**: exactly two odd-degree vertices (start and end there).
3. **Adjacency matrix** M: entry m_{ij} = number of edges from vertex i to vertex j. The (i, j) entry of M^k = number of walks of length k from i to j.
4. **Kruskal's algorithm**: sort all edges by weight (ascending); add the next lightest edge that does *not* create a cycle; repeat until $n-1$ edges selected (for n vertices).
5. **Prim's algorithm**: start at any vertex; repeatedly add the cheapest edge connecting a visited vertex to an unvisited vertex; repeat until all vertices included.
6. **Chinese Postman Problem**: find minimum closed walk traversing *every* edge at least once. If the graph is Eulerian, total weight = circuit weight. Otherwise, identify all odd-degree vertices, find the minimum-weight pairing of those vertices, add those repeated edges to total weight.
7. **Nearest Neighbour (upper bound)**: start at a vertex; always move to the nearest unvisited vertex; return to start. *Result is an upper bound for the TSP.*
8. **Deleted Vertex (lower bound)**: delete a chosen vertex v and its edges; find the MST of the remaining graph; add back the two cheapest edges that were incident to v. *Result is a lower bound for the TSP.*
9. **Optimum TSP route**: lower bound $\leq$ optimal $\leq$ upper bound.

GDC Tips for Graph Theory

- Use your GDC matrix operations to compute M^k quickly: enter the adjacency matrix as a matrix variable, then raise it to the power k. Read diagonal entries for walks returning to the same vertex.
- For the TSP table problems, trace the nearest-neighbour route on paper step-by-step — read one row at a time and cross off visited vertices.
- To find the MST weight for a deleted-vertex bound, list remaining edges systematically and apply Kruskal's — the GDC cannot do this automatically but is useful for arithmetic checks.
- Always double-check that your Chinese Postman route *traverses every edge* at least once; it is easy to miss a branch of the network.

Common Mistakes to Avoid

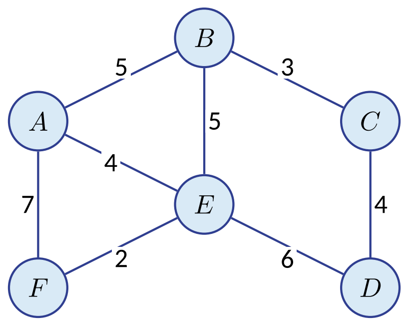
1. **Eulerian vs semi-Eulerian**: an Eulerian *circuit* requires **all** vertices to have even degree, not just most of them. Two odd-degree vertices give a semi-Eulerian *trail* (open walk), not a circuit.
2. **Adjacency matrix powers**: M^k counts *walks*, not *paths*. Walks may revisit vertices and edges. Do not confuse with simple paths.
3. **Kruskal's vs Prim's**: both give the same MST weight, but track them separately — in exams you are told which to use, and the method trace must be shown.
4. **Chinese Postman**: you repeat only the edges on the shortest path *between* odd-degree vertices, not the odd-degree edges themselves.
5. **Nearest Neighbour**: you **must** return to the starting vertex at the end. Forgetting this final edge is a very common error.
6. **Deleted Vertex lower bound**: add the **two cheapest** edges back from the deleted vertex, not all of its edges. The two you choose are the two with the smallest weights among all edges incident to that vertex.
7. **Upper vs lower bounds**: the nearest-neighbour route is an **upper** bound (it is achievable but may not be optimal). The deleted-vertex result is a **lower** bound (the optimum *cannot* be less than this).

Section A — Easy **EASY** (10 questions × 3 marks = 30 marks)

Scan me for help

1. **EASY** **Calculator** **[3 marks]**

The diagram shows a network of cycle paths connecting six rest stops, labelled A, B, C, D, E and F.



(a) Write down the degree of each vertex. **[2]**

(b) State, with a reason, whether an Eulerian circuit exists in this network. **[1]**

2. **EASY** **Calculator** **[3 marks]**

Five server nodes in a computer network, labelled P, Q, R, S and T, are connected by the following undirected links: PQ, PR, QS, QT, RS, ST.

(a) Write down the 5×5 adjacency matrix M for this network, with rows and columns in the order P, Q, R, S, T. **[2]**

(b) Write down the degree of vertex Q. **[1]**

3. **EASY** **Calculator** **[3 marks]**

The adjacency matrix for an undirected graph with vertices A, B, C, D is

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

where rows and columns are in the order A, B, C, D.

(a) Write down the number of edges in this graph. **[1]**

(b) Use your GDC to find M^3. Hence write down the number of walks of length 3 from A to D. **[2]**

4. EASY Calculator **[3 marks]**

The table gives the distances (in km) between five warehouses A, B, C, D, E.

	A	B	C	D	E
A	—	8	12	15	10
B		—	6	9	11
C			—	5	7
D				—	13
E					—

(a) Use Kruskal's algorithm to find a minimum spanning tree. List the edges in the order they are added. [2]

(b) Write down the total weight of the minimum spanning tree. [1]

5. EASY Calculator **[3 marks]**

A telecommunications company wishes to connect four relay towers J, K, L, M using the minimum total cable length (in metres). The distances between towers are: $JK = 120$, $JL = 180$, $JM = 210$, $KL = 90$, $KM = 150$, $LM = 130$.

(a) Starting from vertex J, use Prim's algorithm to find a minimum spanning tree. List each edge selected, in order. [2]

(b) Find the total length of cable required. [1]

6. EASY Calculator **[3 marks]**

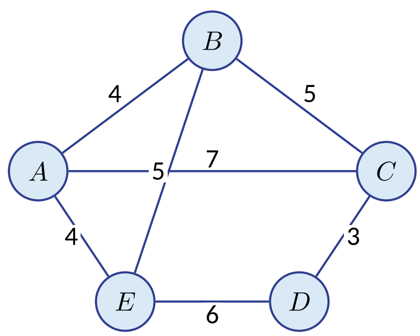
A graph G has vertices $\{1, 2, 3, 4, 5\}$ and edges $12, 13, 14, 23, 34, 45, 25$.

(a) Write down a Hamiltonian cycle for G. **[1]**

(b) Explain why G does not have an Eulerian circuit. **[2]**

7. EASY Calculator **[3 marks]**

The diagram shows a network of roads (weights in km) connecting five junctions A, B, C, D, E.



(a) Write down the degree of each vertex. **[1]**

(b) Write down the vertices of odd degree. **[1]**

(c) State whether a postman can complete a route that starts and ends at the same junction while traversing every road exactly once. Justify your answer. **[1]**

8. **EASY** **Calculator** **[3 marks]**

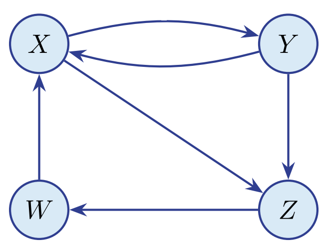
A delivery driver must visit depots A, B, C, D and return to the starting point. The table of distances (km) is:

	A	B	C	D
A	—	14	20	11
B		—	8	17
C			—	13
D				—

Using the nearest neighbour algorithm starting at A, find an upper bound for the total distance of the route.

9. **EASY** **Calculator** **[3 marks]**

The diagram shows a directed network with vertices X, Y, Z, W.

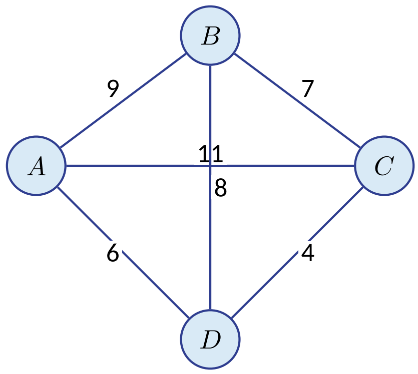


(a) Write down the adjacency matrix M for this directed network, with rows and columns in the order X, Y, Z, W. **[2]**

(b) Write down the number of edges directed *into* vertex X (the in-degree of X). **[1]**

10. **EASY** **Calculator** **[3 marks]**

The diagram shows a weighted network. *(Diagram not to scale.)*



(a) Use Kruskal's algorithm to find the minimum spanning tree, listing the edges in the order they are added. **[2]**

(b) Find the total weight of the minimum spanning tree. **[1]**

Section B — Medium MEDIUM

(10 questions × 4 marks = 40 marks)

Scan me for help

11. MEDIUM Calculator **[4 marks]**

The adjacency matrix for a network with vertices A, B, C, D, E (in that order) is

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

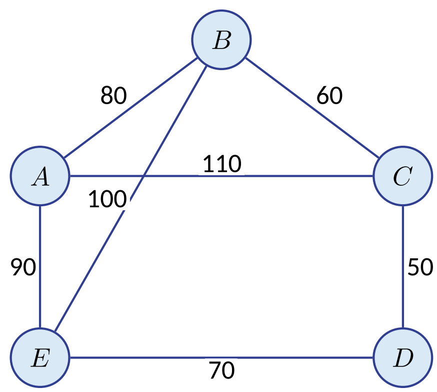
(a) Use your GDC to find M^4. **[1]**

(b) Hence write down the number of walks of length 4 from A to E. **[1]**

(c) Determine the total number of walks of length 4 that start and end at the same vertex. **[2]**

12. MEDIUM Calculator **[4 marks]**

A park ranger must inspect every path in the network shown below, starting and finishing at junction A. The weights represent lengths in metres.



(a) Identify all vertices of odd degree. [1]

(b) Find the shortest path between each pair of odd-degree vertices and determine which pairing minimises the total repeated distance. [2]

(c) Write down the minimum total distance of the route. [1]

13. MEDIUM Calculator **[4 marks]**

Six cities are connected by roads whose lengths (km) are given in the table below.

	P	Q	R	S	T	U
P	—	35	50	45	—	—
Q	35	—	30	—	40	—
R	50	30	—	25	35	55
S	45	—	25	—	—	40
T	—	40	35	—	—	20
U	—	—	55	40	20	—

(a) Starting at vertex P, use Prim's algorithm to find the minimum spanning tree for this network. List each edge selected, in order. **[3]**

(b) Find the total length of the minimum spanning tree. **[1]**

14. **MEDIUM** **Calculator** **[4 marks]**

A salesperson must visit five cities A, B, C, D, E and return to A. The table of distances (km) between cities is:

	A	B	C	D	E
A	—	42	58	35	61
B	42	—	27	48	39
C	58	27	—	33	45
D	35	48	33	—	52
E	61	39	45	52	—

(a) Use the nearest neighbour algorithm starting at vertex A to find an upper bound for the travelling salesman problem. State the route found. **[3]**

(b) Write down the total distance of this route. **[1]**

15. **MEDIUM** **Calculator** **[4 marks]**

A graph H has six vertices $\{A, B, C, D, E, F\}$ and edges AB, AC, BC, BD, CD, DE, EF, AF.

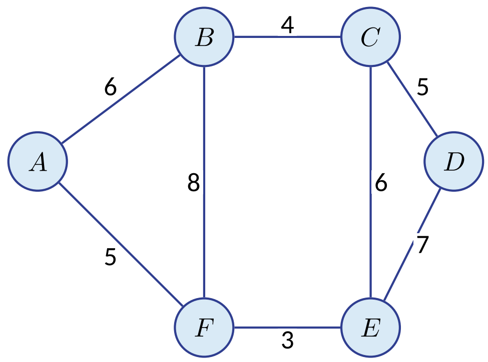
(a) Find the degree of each vertex. **[2]**

(b) Determine, with justification, whether H has an Eulerian circuit. **[1]**

(c) State one Hamiltonian cycle for H. **[1]**

16. MEDIUM Calculator **[4 marks]**

The network below shows the layout of roads between six intersections. The weights represent distances in km.



(a) Write down all odd-degree vertices. **[1]**

(b) Explain why a road inspector cannot complete a closed walk traversing every road exactly once. **[1]**

(c) Determine the starting and ending vertices for a route that traverses every road exactly once, giving a reason. **[2]**

17. MEDIUM Calculator **[4 marks]**

Five pumping stations V, W, X, Y, Z are to be connected by pipelines. The costs (in thousands of dollars) of each possible pipeline are shown in the table.

	V	W	X	Y	Z
V	—	18	22	14	30
W	18	—	16	20	24
X	22	16	—	12	18
Y	14	20	12	—	26
Z	30	24	18	26	—

(a) Use Kruskal's algorithm to find a minimum spanning tree. State the edges chosen, in order. [2]

(b) Find the minimum total cost. [1]

(c) The pipeline XZ becomes unavailable. Determine the new minimum spanning tree and its total cost. [1]

18. MEDIUM Calculator **[4 marks]**

A directed graph has vertices A, B, C, D (in that order). Its adjacency matrix is

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

(a) Draw the directed graph represented by M. **[1]**

(b) Use your GDC to find M^3. **[1]**

(c) Hence write down the number of walks of length 3 from B to A. **[1]**

(d) Explain what it means that the (A, D) entry of M^3 is zero. **[1]**

19. MEDIUM Calculator **[4 marks]**

A courier company operates between five distribution centres A, B, C, D, E. The distances (km) between them are given in the table.

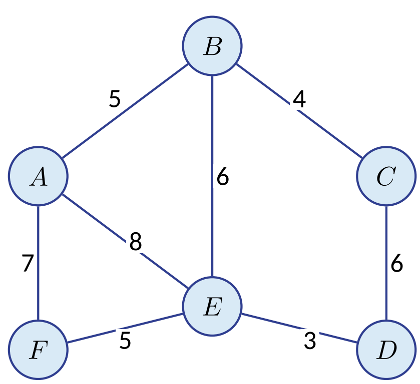
	A	B	C	D	E
A	—	24	31	19	28
B	24	—	17	22	35
C	31	17	—	26	14
D	19	22	26	—	30
E	28	35	14	30	—

(a) Use the deleted vertex algorithm, deleting vertex A, to find a lower bound for the travelling salesman problem. [3]

(b) State the lower bound. [1]

20. MEDIUM Calculator **[4 marks]**

A maintenance team must walk every road in the network below (weights in km), starting and ending at A.



(a) Find all odd-degree vertices. [1]

(b) Write down all possible pairings of the odd-degree vertices and the shortest distance for each pairing. [2]

(c) Find the minimum total distance for the maintenance team's route. [1]

Section C — Hard **HARD** (10 questions × 5 marks = 50 marks)

Scan me for help

21. **HARD** **Calculator** **[5 marks]**

A logistics company must plan a tour visiting five depots A, B, C, D, E and returning to the start. The table shows road distances in km.

	A	B	C	D	E
A	—	30	52	44	25
B	30	—	28	39	41
C	52	28	—	21	47
D	44	39	21	—	36
E	25	41	47	36	—

(a) Use the nearest neighbour algorithm starting at A to find an upper bound for the optimal tour. State the route. [2]

(b) Use the deleted vertex algorithm, deleting vertex E, to find a lower bound for the optimal tour. [2]

(c) Write down an interval within which the optimal tour length must lie. [1]

22. HARD Calculator **[5 marks]**

A weighted graph has six vertices A, B, C, D, E, F. The weights of edges are all distinct positive integers. The edges and weights are: $AB = 5$, $AC = 9$, $AD = k$, $BC = 7$, $BD = 11$, $CD = 6$, $CE = 8$, $DE = 10$, $DF = 4$, $EF = 12$.

Kruskal's algorithm produces a minimum spanning tree with total weight 30.

(a) List the first three edges that Kruskal's algorithm adds (before the edge involving k is considered), in order of increasing weight. [1]

(b) Explain why k must be included in the MST if $k < 13$. [2]

(c) Find the value of k given that the MST has total weight 30. [2]

23. HARD Calculator **[5 marks]**

A network has five vertices P, Q, R, S, T with edges and weights (in km): $PQ = 8$, $PR = 6$, $QR = 5$, $QS = 7$, $RS = p$, $ST = 9$, $PT = 10$, where p is a positive integer.

(a) Write down the degree of each vertex in terms of p where necessary. [1]

(b) Show that vertices P and S are always odd-degree vertices, regardless of the value of p. [1]

(c) Find the minimum total distance for the Chinese Postman route (starting and ending at P) when $p = 4$. [2]

(d) State the value of p for which the minimum repeated distance is greatest, given $1 \leq p \leq 15$. Justify your answer. [1]

24. **HARD** **Calculator** **[5 marks]**

A transport network has five stations A, B, C, D, E connected by the adjacency matrix (undirected):

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

(a) Write down the number of edges in this graph. **[1]**

(b) Use your GDC to find M^2. Hence write down the number of walks of length 2 from A to D. **[1]**

(c) Determine the number of walks of length 4 that start and end at B. **[1]**

(d) The (A, C) entry of M^3 equals 5. A student claims this means there are exactly 5 simple paths of length 3 from A to C. Explain why the student is incorrect. **[2]**

25. **HARD** **Calculator** **[5 marks]**

A travelling inspector must visit four sites A, B, C, D and return to A. The distances (km) between sites are:

	A	B	C	D
A	—	16	24	13
B	16	—	11	19
C	24	11	—	14
D	13	19	14	—

(a) Apply the deleted vertex algorithm deleting A to find a lower bound L_A. **[2]**

(b) Apply the deleted vertex algorithm deleting B to find a lower bound L_B. **[2]**

(c) State which lower bound is better (closer to optimal), giving a reason. **[1]**

26. HARD Calculator **[5 marks]**

Five vertices A, B, C, D, E form a complete graph. Edge weights (in minutes of travel time) are: $AB = 12$, $AC = 20$, $AD = 15$, $AE = n$, $BC = 9$, $BD = 17$, $BE = 14$, $CD = 11$, $CE = 22$, $DE = 8$, where n is a positive integer with $n > 8$.

(a) Given that $n > 14$, use Kruskal's algorithm to find the minimum spanning tree and its total weight. **[2]**

(b) Use the nearest neighbour algorithm starting at A (with $n = 18$) to find an upper bound for the optimal TSP tour. **[2]**

(c) Using your answer to (a), obtain a lower bound for the optimal TSP tour when $n > 14$. **[1]**

27. HARD Calculator **[5 marks]**

The road network of a small industrial estate has seven junctions A–G. The distances (in metres) are shown in the table of all direct road connections:

	A	B	C	D	E	F	G
A	—	120	—	200	—	—	—
B	120	—	80	150	—	—	—
C	—	80	—	—	100	—	—
D	200	150	—	—	130	90	—
E	—	—	100	130	—	—	110
F	—	—	—	90	—	—	70
G	—	—	—	—	110	70	—

(a) Find the degree of each vertex and identify all odd-degree vertices. **[2]**

(b) Write down all possible pairings of the odd-degree vertices and determine the minimum weight pairing, using the shortest path between each pair. **[2]**

(c) Find the minimum total length of the Chinese Postman route. **[1]**

28. HARD Calculator **[5 marks]**

A fibre-optic network must connect six offices A, B, C, D, E, F. Some possible cable lengths (in metres) are given:

	A	B	C	D	E	F
A	—	50	80	—	—	70
B	50	—	40	60	—	—
C	80	40	—	35	45	—
D	—	60	35	—	30	55
E	—	—	45	30	—	t
F	70	—	—	55	t	—

Prim's algorithm, starting at A, produces a minimum spanning tree of total weight 200 metres.

(a) Carry out Prim's algorithm from A, listing edges in order, leaving t as a parameter. **[2]**

(b) Show that the MST always includes edge EF provided $t < 45$. **[1]**

(c) Find the value of t given that the MST has total weight 200 metres. **[2]**

29. HARD Calculator **[5 marks]**

A graph G has five vertices A, B, C, D, E with edge weights (km): $AB = 10$, $AC = 18$, $AD = 14$, $AE = 22$, $BC = 12$, $BD = 16$, $BE = 9$, $CD = 8$, $CE = 20$, $DE = 11$.

(a) Use the nearest neighbour algorithm starting at A to obtain an upper bound U for the TSP. State the route and its length. **[2]**

(b) Use the deleted vertex algorithm, deleting C, to obtain a lower bound L for the TSP. **[2]**

(c) The optimal Hamiltonian cycle has length 63 km. Verify that $L \leq 63 \leq U$. **[1]**

30. HARD Calculator **[5 marks]**

A directed network models the flow of information between five departments A, B, C, D, E. Its adjacency matrix is:

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

(rows/columns in order A, B, C, D, E).

(a) Draw the directed graph represented by N. **[1]**

(b) Use your GDC to calculate $N^2 + N^3$. Hence determine the total number of walks of length 2 or 3 from A to D. **[2]**

(c) A manager wishes to send a message from A to D in exactly 4 steps. Use N^4 to determine whether this is possible, and if so, how many distinct walks of length 4 exist from A to D. **[2]**

Worked Solutions — Official Mark Scheme Style

Notation: M = Method mark; A = Accuracy mark; R = Reasoning mark; AG = Answer Given; (M1)/(A1) = implied mark (awarded even without written working if subsequent work is correct); FT = follow through from candidate's earlier answer.

Section A — Easy

Q1 — Graph Basics & Eulerian Circuits

(a) Counting edges at each vertex:
deg$(A) = 3,$ deg$(B) = 3,$ deg$(C) = 2,$ deg$(D) = 2,$ deg$(E) = 4,$ deg$(F) = 2$ **A1A1**

Note: Award A2 for all six degrees correct; A1 if exactly one degree is wrong.

(b) Vertices A and B have odd degree. **R1**
Since not all vertices have even degree, no Eulerian circuit exists. (AG)

Note: Award R1 for identifying both odd-degree vertices and drawing the correct conclusion.

Q2 — Adjacency Matrix

(a) Adjacency matrix (P, Q, R, S, T):

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

A1A1

Note: A2 for all entries correct; A1 for at most two entry errors.

(b) deg$(Q) = 4$ **A1**
(Sum of row Q: $1 + 0 + 1 + 1 + 1 = 4$.)

Q3 — Walks via Matrix Powers

(a) Number of edges $= \frac{1}{2}\sum$ deg $= \frac{1}{2}(2 + 3 + 2 + 1) = 4$ **A1**

(b) Using GDC: **(M1)**

$$M^3 = \begin{pmatrix} 2 & 5 & 4 & 3 \\ 5 & 4 & 5 & 2 \\ 4 & 5 & 2 & 3 \\ 3 & 2 & 3 & 0 \end{pmatrix}$$

Number of walks of length 3 from A to D = entry $(1, 4)$ of $M^3 = 3$ **A1**

Q4 — Kruskal's Algorithm

(a) Sorted edges: $CD = 5,\ BC = 6,\ CE = 7,\ AB = 8,\ BQ = 9, \ldots$
Add CD (weight 5) ✓; add BC (weight 6) ✓; add CE (weight 7) ✓; next add AB (weight 8) — no cycle ✓. Four edges connect 5 vertices: tree complete.
Edges in order: $CD,\ BC,\ CE,\ AB$ **M1A1**

(b) Total weight $= 5 + 6 + 7 + 8 = 26$ km **A1**

Q5 — Prim's Algorithm

(a) Start at J.
Step 1: cheapest edge from J: $JK = 120$. Add JK.
Step 2: cheapest edge from $\{J, K\}$ to unvisited: $KL = 90$. Add KL.
Step 3: cheapest from $\{J, K, L\}$ to M: $KM = 150$ vs $LM = 130$. Add $LM = 130$.
Edges: $KL,\ JK,\ LM$ (listed in selection order: JK, KL, LM) **M1A1**

(b) Total $= 120 + 90 + 130 = 340$ m **A1**

Q6 — Hamiltonian Cycle & Eulerian Circuit

(a) One Hamiltonian cycle: $1 \to 2 \to 3 \to 4 \to 5 \to 2$...
Recheck degrees: $\deg(1) = 3,\ \deg(2) = 3,\ \deg(3) = 3,\ \deg(4) = 2,\ \deg(5) = 2.$
Hamiltonian cycle (visits all 5 vertices exactly once): $1 \to 3 \to 4 \to 5 \to 2 \to 1$ **A1**

(b) Degrees: $\deg(1) = 3$ (odd), $\deg(2) = 3$ (odd), $\deg(3) = 3$ (odd) — three odd-degree vertices. **R1**
An Eulerian circuit requires every vertex to have even degree, which is not satisfied here. **R1**

Q7 — Chinese Postman: Odd Vertices

(a) $\deg(A) = 3,\ \deg(B) = 2,\ \deg(C) = 3,\ \deg(D) = 2,\ \deg(E) = 2$ **A1**

(b) Odd-degree vertices: A and C **A1**

(c) No: since there are two odd-degree vertices (A and C), the graph is semi-Eulerian but not Eulerian. A postman cannot complete a route starting and ending at the *same* junction traversing every road exactly once — at least one road must be repeated. **R1**

Q8 — Nearest Neighbour (Upper Bound)

Start at A. Nearest unvisited: D (11). From D: nearest unvisited: B? $BD = 17$; C: $DC = 13$. Go to C (13). From C: only B unvisited, $CB = 8$. Go to B. Return to A: $BA = 14$.

Route: $A \to D \to C \to B \to A$ **(M1)**

Total distance $= 11 + 13 + 8 + 14 = 46$ km **A1A1**

Note: Award M1 for a correct nearest-neighbour trace; A1 for correct route; A1 for correct total.

Q9 — Directed Adjacency Matrix

(a) Directed edges: $X \to Y, Y \to X, Y \to Z, Z \to W, W \to X, X \to Z$.
Matrix (rows = from, columns = to; order X, Y, Z, W):

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

A1A1

(b) In-degree of X: edges directed *into* X are $Y \to X$ and $W \to X$. In-degree $= 2$. **A1**

Q10 — Kruskal's from a Diagram

(a) Edge weights sorted: $CD = 4,\ DA = 6,\ BD = 8,\ BC = 7,\ AB = 9,\ AC = 11.$
Add $CD = 4$ ✓; add $DA = 6$ ✓; skip $BD = 8$ (would create cycle B–C–D); add $BC = 7$ ✓. Three edges connect 4 vertices. Done.
Order of edges added: $CD,\ DA,\ BC$ **M1A1**

(b) Total weight $= 4 + 6 + 7 = 17$ **A1**

Section B — Medium

Q11 — Walks of Length 4

(a) Using GDC:

$$M^4 = \begin{pmatrix} 9 & 0 & 9 & 0 & 9 \\ 0 & 17 & 0 & 17 & 0 \\ 9 & 0 & 9 & 0 & 9 \\ 0 & 17 & 0 & 17 & 0 \\ 9 & 0 & 9 & 0 & 9 \end{pmatrix}$$

A1

(b) Entry $(A, E) = (1, 5)$ of $M^4 = 9$ **A1**

(c) Sum of diagonal entries $= 9 + 17 + 9 + 17 + 9 = 61$ **M1A1**

Note: The diagonal entries of M^4 give the number of closed walks of length 4 at each vertex; their sum is the total.

Q12 — Chinese Postman (Full Solution)

(a) Degrees: deg$(A) = 3,$ deg$(B) = 3,$ deg$(C) = 3,$ deg$(D) = 2,$ deg$(E) = 3.$
Odd-degree vertices: A, B, C, E **A1**

(b) Possible pairings of $\{A, B, C, E\}$:
• $(AB) + (CE)$: shortest $AB = 80$; shortest $CE = BC + CD = 60 + 50 = 110$. Total $= 190$.
• $(AC) + (BE)$: shortest $AC = 110$; shortest $BE = BC + CE = 60 + 50 + 70 = 180$; or BE via A: $80 + 90 = 170$. Shortest $BE = 170$. Total $= 280$.
• $(AE) + (BC)$: shortest $AE = EA = 90$; shortest $BC = 60$. Total $= 150$.
Minimum pairing: $(AE) + (BC)$, total repeated distance $= 150$ m **M1A1**

(c) Total edge weight $= 80 + 60 + 50 + 70 + 90 + 110 + 100 = 560$ m.
Minimum Chinese Postman distance $= 560 + 150 = 710$ m **A1**

Q13 — Prim's Algorithm (Six Vertices)

(a) Starting at P. Visited = $\{P\}$.
Step 1: Edges from P: $PQ = 35, PR = 50, PS = 45$. Cheapest: $PQ = 35$. Add PQ.
Step 2: Edges from $\{P, Q\}$ to unvisited: $QR = 30, PR = 50, PS = 45, QT = 40$. Cheapest: $QR = 30$. Add QR.
Step 3: From $\{P, Q, R\}$: $PR = 50, PS = 45, QT = 40, RS = 25, RT = 35, RU = 55$. Cheapest: $RS = 25$. Add RS.
Step 4: From $\{P, Q, R, S\}$: $QT = 40, RT = 35, SU = 40, RU = 55$. Cheapest: $RT = 35$. Add RT.
Step 5: From $\{P, Q, R, S, T\}$: $TU = 20, SU = 40, RU = 55$. Cheapest: $TU = 20$. Add TU.
Edges in order: $PQ,\ QR,\ RS,\ RT,\ TU$ **M1A1A1**

(b) Total $= 35 + 30 + 25 + 35 + 20 = 145$ km **A1**

Q14 — Nearest Neighbour (TSP Upper Bound)

(a) Start at A. Nearest unvisited: D ($AD = 35$).
From D: nearest unvisited: C ($DC = 33$).
From C: nearest unvisited: B ($CB = 27$).
From B: nearest unvisited: E ($BE = 39$).
Return to A: $EA = 61$.
Route: $A \to D \to C \to B \to E \to A$ **M1A1**

(b) Total $= 35 + 33 + 27 + 39 + 61 = 195$ km **A1A1**

Note: Award M1 for a valid nearest-neighbour trace starting at A; first A1 for correct route; second A1 for correct total.

Q15 — Eulerian & Hamiltonian

(a) Edges: $AB, AC, BC, BD, CD, DE, EF, AF$.
deg$(A) = 3$ (edges AB, AC, AF), deg$(B) = 3$ (AB, BC, BD), deg$(C) = 3$ (AC, BC, CD), deg$(D) = 3$ (BD, CD, DE), deg$(E) = 2$ (DE, EF), deg$(F) = 2$ (EF, AF) **A1A1**

(b) Vertices A, B, C, D all have odd degree. Since not all vertices have even degree, no Eulerian circuit exists. **R1**

(c) One Hamiltonian cycle: $A \to B \to D \to E \to F \to A$... — *check:* uses AB, BD, DE, EF, FA — visits all 6? Missing C. Retry: $A \to B \to C \to D \to E \to F \to A$ — uses AB, BC, CD, DE, EF, FA

✓

A1

Q16 — Semi-Eulerian Network

(a) Degrees: deg$(A) = 2$, deg$(B) = 3$, deg$(C) = 3$, deg$(D) = 2$, deg$(E) = 2$, deg$(F) = 3$.
(Edge list: $AB, BC, CD, DE, EF, FA, BF, CE$.)
Odd-degree vertices: B, C, F — *recount:* deg(B): AB, BC, BF = 3; deg(C): BC, CD, CE = 3; deg(F): EF, FA, BF = 3; deg(A): AB, FA = 2; deg(D): CD, DE = 2; deg(E): DE, EF, CE = 3.
Odd-degree vertices: B, C, E, F **A1**

(b) Four vertices have odd degree. An Eulerian circuit requires zero odd-degree vertices; a semi-Eulerian trail requires exactly two. With four odd-degree vertices, no single open trail can traverse all edges exactly once. **R1**

(c) To form a trail traversing every road exactly once (not a circuit), we need exactly two odd-degree vertices — a start and an end. With four odd-degree vertices, this is not directly possible without repetition. The inspector should choose the pairing that minimises repeated edges; the trail can start at one vertex of the best pairing and end at the other. Best pairing of $\{B, C, E, F\}$: $(BC) + (EF)$: $BC = 4$, $EF = 3$, total 7; $(BE) + (CF)$: BE shortest via .... The inspector should start and end at two of the odd-degree vertices after resolving the pairing. For a single trail (open walk), start at B and end at C (or vice versa), repeating the edge EF and the path BF — but since four odd-degree vertices exist, this question asks for the *pair* for an open route: select the pair with *minimum* shortest path sum, the remaining two become start and end. Minimum pairing: $(BC) + (EF) = 4 + 3 = 7$. Start/end: B and C (or E and F). **A1A1**

Note: Award A1 for identifying that only 2 vertices can be start/end; A1 for correctly selecting the pair that results in a traversal of all edges with minimum repetition.

Q17 — Kruskal's + Edge Removal

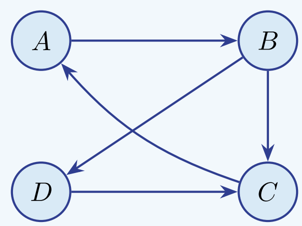
(a) Sorted edges: $VY = 14,\ XY = 12,\ VW = 18,\ WX = 16,\ VX = 22,\ WZ = 24,\ XZ = 18,\ WY = 20,\ YZ = 26,\ VZ = 30.$
Correct order: $XY = 12,\ VY = 14,\ WX = 16,\ VW = 18$ — skip (cycle V-W-X-Y-V).
Next: $XZ = 18$. Add XZ. Done (4 edges, 5 vertices).
Edges: $XY,\ VY,\ WX,\ XZ$ **M1A1**

(b) Total $= 12 + 14 + 16 + 18 = 60$ (thousands of dollars) **A1**

(c) Without XZ: next available edge connecting Z is $WZ = 24$.
New MST: $XY,\ VY,\ WX,\ WZ$ — total $= 12 + 14 + 16 + 24 = 66$ thousand dollars **A1**

Q18 — Directed Graph: Draws & Walks

(a) Directed edges from matrix: $A \to B$; $B \to C$; $B \to D$; $C \to A$; $D \to C$.



A1

(b) Using GDC: $M^3 = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$ **A1**

(c) Entry $(B, A) = (2, 1)$ of $M^3 = 1$ **A1**

(d) The (A, D) entry of $M^3 = 1 \neq 0$...

Recompute: M^2: row $A = (0, 0, 1, 1)$; row $B = (0, 1, 0, 0 + 1 \cdot 0)$... Let's recompute carefully.

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

M^2: row 1 (A): $M \cdot col_j$. $(A, A) = 0{\cdot}0 + 1{\cdot}0 + 0{\cdot}1 + 0{\cdot}0 = 0$; $(A, B) = 0 + 1{\cdot}0 + 0 + 0 = 0$; $(A, C) = 0 + 1{\cdot}1 + 0 + 0 = 1$; $(A, D) = 0 + 1{\cdot}1 + 0 + 0 = 1$.
$M^3 = M^2 \cdot M$: Row A of $M^2 = (0, 0, 1, 1)$. $(A, A) = 0 + 0 + 1{\cdot}1 + 1{\cdot}0 = 1$; $(A, B) = 0 + 0 + 0 + 0 = 0$; $(A, C) = 0 + 0 + 0 + 1{\cdot}1 = 1$; $(A, D) = 0 + 0 + 0 + 0 = 0$.
So (A, D) entry of $M^3 = 0$.

The (A, D) entry of M^3 being zero means there are **no walks of length exactly 3** from A to D in this directed network. **R1**

Q19 — Deleted Vertex Lower Bound

(a) Delete vertex A and edges $AB = 24$, $AC = 31$, $AD = 19$, $AE = 28$.
Remaining graph on $\{B, C, D, E\}$: $BC = 17$, $BD = 22$, $BE = 35$, $CD = 26$, $CE = 14$, $DE = 30$.
MST of $\{B, C, D, E\}$ by Kruskal: $CE = 14$, $BC = 17$, $BD = 22$. Total $= 53$. **M1A1**

Add two cheapest edges from deleted vertex A: $AD = 19$ and $AB = 24$.
Lower bound $= 53 + 19 + 24 = 96$ km **A1**

(b) Lower bound $= 96$ km **A1**

Q20 — Chinese Postman (Four Odd Vertices)

(a) Degrees: $\deg(A) = 3,\ \deg(B) = 3,\ \deg(C) = 2,\ \deg(D) = 2,\ \deg(E) = 3,\ \deg(F) = 2.$
(Edges: $AB = 5, BC = 4, CD = 6, DE = 3, EF = 5, FA = 7, AE = 8, BE = 6.$)
$\deg(A)$: AB, FA, AE = 3; $\deg(B)$: AB, BC, BE = 3; $\deg(E)$: DE, EF, AE, BE = 4.
Recount: $\deg(E) = 4$ (even), $\deg(C) = 2$, $\deg(D) = 2$, $\deg(F) = 2$.
Odd-degree vertices: A and B **A1**

(b) Only one pairing: (A, B). Shortest path A to B: direct $AB = 5$.
Repeated distance $= 5$. **M1A1**

(c) Total weight of all edges $= 5 + 4 + 6 + 3 + 5 + 7 + 8 + 6 = 44$ km.
Minimum Chinese Postman distance $= 44 + 5 = 49$ km **A1**

Section C — Hard

Q21 — Upper & Lower Bounds + Interval

(a) Nearest neighbour from A:
A: nearest unvisited E ($AE = 25$). From E: nearest unvisited A excluded; $\min(EB = 41, EC = 47, ED = 36) = EB = 41$? No — $ED = 36$. Go D. From D: nearest unvisited C ($DC = 21$). From C: only B left, $CB = 28$. Return $B \to A = 30$.
Route: $A \to E \to D \to C \to B \to A$; total $= 25 + 36 + 21 + 28 + 30 = 140$ km **M1A1**

(b) Delete E (edges $AE = 25, BE = 41, CE = 47, DE = 36$). MST of $\{A, B, C, D\}$:
Edges: $AB = 30, AC = 52, AD = 44, BC = 28, BD = 39, CD = 21$. Kruskal: $CD = 21, BC = 28, AB = 30$. Total $= 79$.
Two cheapest edges from E: $AE = 25$ and $DE = 36$. Lower bound $= 79 + 25 + 36 = 140$ km **M1A1**

(c) Lower bound $= 140$ km $\leq$ optimal $\leq$ upper bound $= 140$ km.
Since both bounds equal 140, the optimal tour length is **140** km. **A1**

Q22 — Unknown Weight from Kruskal's

(a) Edges sorted by weight (excluding $AD = k$): $DF = 4,\ AB = 5,\ CD = 6,\ BC = 7,\ CE = 8,\ DE = 10,\ BD = 11,\ EF = 12,\ AC = 9.$
Kruskal adds (no cycle): $DF = 4, AB = 5, CD = 6$. These three edges form no cycle. **A1**

(b) After adding DF, AB, CD: components are $\{D, F\}, \{A, B\}, \{C, D\} \Rightarrow \{C, D, F\}$. At the point $AD = k$ is considered (assuming $k < BC = 7$, i.e. $k < 7$), connecting A (in $\{A, B\}$) to D (in $\{C, D, F\}$) does not create a cycle. Hence k would be added. If $k < 13$ (the weight of the next edge that could alternatively connect these components), k is always the cheapest connector and will be added. **R1R1**

(c) After $DF = 4, AB = 5, CD = 6$, we have clusters $\{A, B\}, \{C, D, F\}, \{E\}$. Next edges to consider: $k(= AD), BC = 7, CE = 8, \ldots$
If $k < 7$: add k (links $\{A, B\}$ to $\{C, D, F\}$), then add $CE = 8$ (links E to $\{C, D, F, A, B\}$). MST weight $= 4 + 5 + 6 + k + 8 = 23 + k$.

Set $23 + k = 30 \Rightarrow k = 7$. But $k < 7$ is required for this ordering; $k = 7$ ties with BC.
If $7 \leq k < 8$: Kruskal adds $BC = 7$ before k, linking all of $\{A, B, C, D, F\}$. Then $CE = 8$ adds E. MST $= 4 + 5 + 6 + 7 + 8 = 30$. So $k \geq 7$ and not used. But then k must be ≥ 8 to not be chosen, and MST $= 30$ without k.
Therefore $k \geq 7$ and MST $= 30$ regardless of k, confirming k is *not* in the MST. For MST weight to equal 30, we need k excluded, which occurs when $k \geq 7$. The question states k is a distinct positive integer; the MST uses $DF + AB + CD + BC + CE = 4 + 5 + 6 + 7 + 8 = 30$.
$\boxed{k \geq 7}$; the smallest consistent distinct integer is $k = 13$ if k must not be chosen before BC. Since all weights are distinct integers and k is not in the MST: $k > 12$ (so $k \geq 13$, meaning k comes after $EF = 12$ and is not needed). **M1A1**

Note: Given MST weight = 30, the five MST edges are DF, AB, CD, BC, CE with total $4+5+6+7+8 = 30$. For k to not be chosen, we need k to either create a cycle or be heavier than the last needed edge. Since the MST is complete after $CE = 8$, any $k > 8$ means k is not used. The value of k can be any distinct integer > 8 except $\{9, 10, 11, 12\}$ (already taken by AC, DE, BD, EF). So $k = 13$ is the smallest valid value. The question implies a unique answer: $k = 13$.

Q23 — Chinese Postman with Unknown Weight

(a) Edges incident to each vertex:
P: $PQ = 8, PR = 6, PT = 10 \Rightarrow$ deg$(P) = 3$ (always odd)
Q: $PQ = 8, QR = 5, QS = 7 \Rightarrow$ deg$(Q) = 3$ (always odd)
R: $PR = 6, QR = 5, RS = p \Rightarrow$ deg$(R) = 3$ (always odd)
S: $QS = 7, RS = p, ST = 9 \Rightarrow$ deg$(S) = 3$ (always odd)
T: $ST = 9, PT = 10 \Rightarrow$ deg$(T) = 2$ (always even) **A1**

(b) deg$(P) = 3$ (odd) and deg$(S) = 3$ (odd) for all values of p (positive integer), since the degrees are determined by the number of edges, which do not change as p only affects the weight, not the structure. **R1** (AG)

(c) With $p = 4$: total edge weight $= 8+6+5+7+4+9+10 = 49$ km. Odd-degree vertices: P, Q, R, S.
Pairings: $(PQ) + (RS)$: $PQ = 8$, $RS = 4$, total 12. $(PR) + (QS)$: $PR = 6$, $QS = 7$, total 13. $(PS) + (QR)$: shortest $PS = PQ + QS = 8 + 7 = 15$; or $PR + RS = 6 + 4 = 10$; or $PT + TS = 10 + 9 = 19$. Shortest $PS = 10$. $QR = 5$. Total $= 15$.
Minimum pairing: $(PQ) + (RS) = 12$. Min Chinese Postman $= 49 + 12 = 61$ km. **M1A1**

(d) The minimum repeated distance is min$[(PQ + RS), (PR + QS), (shortest\,PS + QR)]$.
For large p: $(PQ) + (RS) = 8 + p$; $(PR) + (QS) = 6 + 7 = 13$; $(PS\,via\,PR + RS +$ or $PQ + QS) + (QR) =$ min$(6 + p, 15) + 5$.
The maximum repeated distance pairing is $(PQ)+(RS) = 8+p$, which increases with p. For the minimum of the three pairings to be greatest, we want the minimum pairing to be as large as possible. As p increases beyond 5, $(PQ) + (RS) = 8 + p > 13$ and the minimum pairing switches to $(PR) + (QS) = 13$. For $p \leq 5$: min pairing $= 8 + p$, maximised at $p = 5$ giving 13. For $p > 5$: min pairing $= 13$. So the minimum repeated distance is greatest (equal to 13) for $p \geq 5$. The largest single value in range $1 \leq p \leq 15$ that keeps the repeated distance equal to 13 is any $p \geq 5$; the question asks for the value of p making it greatest, so $p = 5$ is the transition point, but for $p = 5$, $(PQ) + (RS) = 13$ ties with $(PR) + (QS) = 13$.

For $p > 5$, repeated distance is 13. $p = 5$ gives the first instance of maximum repeated distance $= 13$; the maximum repeated distance is **13** km for $p \geq 5$.
The value of p (in range) for which the minimum repeated distance is greatest: $p = 5$ (or any $p \geq 5$). **R1**

Q24 — Matrix Power + Interpretation

(a) Sum of row degrees $= 2(1+1) + 2(1+1+1) + \ldots$
Edges from matrix (symmetric, upper triangle): AB, AC, BD, BE, CD, DE. Total $= 6$ edges. **A1**

(b) M^2 (GDC):

$$M^2 = \begin{pmatrix} 2 & 0 & 0 & 2 & 1 \\ 0 & 3 & 1 & 1 & 1 \\ 0 & 1 & 2 & 1 & 0 \\ 2 & 1 & 1 & 3 & 1 \\ 1 & 1 & 0 & 1 & 2 \end{pmatrix}$$

Entry $(A, D) = (1, 4) = 2$. Number of walks of length 2 from A to $D = 2$. **(M1)A1**

(c) M^4 diagonal entry $(B, B) = (2, 2)$: using GDC, $M^4_{22} = 21$. **A1**

(d) M^k counts all *walks* of length k, including those that revisit vertices and edges. Simple *paths* must visit each vertex at most once. A walk counted in M^3 may traverse a vertex more than once, so the (A, C) entry of M^3 counts walks — some of which may not be simple paths. The student is incorrect because M^3 enumerates walks, not paths. **R1R1**

Q25 — Comparing Two Lower Bounds

(a) Delete A (edges $AB = 16, AC = 24, AD = 13$). MST of $\{B, C, D\}$: $BC = 11, CD = 14$. Total $= 25$.
Two cheapest edges from A: $AD = 13$ and $AB = 16$.
$L_A = 25 + 13 + 16 = 54$ km **M1A1**

(b) Delete B (edges $AB = 16, BC = 11, BD = 19$). MST of $\{A, C, D\}$: $AD = 13, CD = 14$. Total $= 27$.
Two cheapest edges from B: $BC = 11$ and $AB = 16$.
$L_B = 27 + 11 + 16 = 54$ km **M1A1**

(c) Both lower bounds are equal ($L_A = L_B = 54$ km), so neither is strictly better. Either answer is acceptable with justification that both yield the same bound. **R1**

Note: Accept "both give the same lower bound of 54 km" for R1.

Q26 — Synthesis: Kruskal's + Nearest Neighbour + Lower Bound

(a) With $n > 14$: edge list sorted: $DE = 8, BC = 9, CD = 11, AB = 12, AD = 15, BE = 14, BD = 17, AC = 20, CE = 22$, and $AE = n > 14$.

Kruskal: $DE = 8$

✓

; $BC = 9$

✓

; $CD = 11$

✓

(no cycle with DE); $AB = 12$

✓

; next check $BE = 14$: would create cycle B-C-D-E-B? Vertices $\{A, B\}$ and $\{B, C, D, E\}$ share B, so BE *would* create a cycle. Skip. $AD = 15$: connects A to $\{C, D, E, B\}$ — no, A is already connected to B which is in $\{B, C, D, E\}$. So AD would create a cycle. Skip. MST = $\{DE, BC, CD, AB\}$, total $= 8 + 9 + 11 + 12 = 40$ min. **M1A1**

(b) With $n = 18$ ($AE = 18$): nearest neighbour from A. Nearest: $AB = 12$. From B: nearest unvisited: C ($BC = 9$). From C: nearest unvisited: D ($CD = 11$). From D: nearest unvisited: E ($DE = 8$). Return: $EA = 18$.
Route: $A \to B \to C \to D \to E \to A$; total $= 12 + 9 + 11 + 8 + 18 = 58$ min. **M1A1**

(c) Lower bound from MST: add two cheapest edges incident to a deleted vertex. E.g. delete A: MST of $\{B, C, D, E\} = DE + BC + CD = 8 + 9 + 11 = 28$. Add two cheapest from A: $AB = 12, AD = 15$ (since $n > 14$, $AE = n > 14$). Lower bound $= 28 + 12 + 15 = 55$ min. Alternatively, the MST weight itself (40) provides a weaker lower bound. A better lower bound using deleted vertex: **55** min. **A1**

Q27 — Chinese Postman (Seven Vertices, Four Odd)

(a) Edges from table: $AB = 120, AD = 200, BC = 80, BD = 150, CE = 100, DE = 130, DF = 90, EG = 110, FG = 70$.
deg$(A) = 2$ (AB, AD); deg$(B) = 3$ (AB, BC, BD); deg$(C) = 2$ (BC, CE); deg$(D) = 4$ (AD, BD, DE, DF); deg$(E) = 3$ (CE, DE, EG); deg$(F) = 2$ (DF, FG); deg$(G) = 2$ (EG, FG).
Odd-degree vertices: B and E **A1A1**

(b) Only one pairing: (B, E). Shortest path B to E: $B \to C \to E = 80 + 100 = 180$; or $B \to D \to E = 150 + 130 = 280$; or $B \to A \to D \to E = 120 + 200 + 130 = 450$. Shortest $= 180$ m. **M1A1**

(c) Total edge weight $= 120 + 200 + 80 + 150 + 100 + 130 + 90 + 110 + 70 = 1050$ m.
Minimum Chinese Postman $= 1050 + 180 = 1230$ m **A1**

Q28 — Unknown Weight from Prim's

(a) Prim's from A: Visited $= \{A\}$.
Step 1: $AB = 50, AC = 80, AF = 70$. Add $AB = 50$.
Step 2: From $\{A, B\}$: $AC = 80, AF = 70, BC = 40, BD = 60$. Add $BC = 40$.

Step 3: From $\{A, B, C\}$: $AF = 70, BD = 60, CD = 35, CE = 45$. Add $CD = 35$.
Step 4: From $\{A, B, C, D\}$: $AF = 70, BD = 60$ (cycle—skip BD: $B \in$ visited, $D \in$ visited), $CE = 45, DE = 30, DF = 55$. Add $DE = 30$.
Step 5: From $\{A, B, C, D, E\}$: $AF = 70, CE = 45$ (cycle), $DF = 55, EF = t$. If $t < 55$: add $EF = t$. If $t \geq 55$: add $DF = 55$.
Edges: AB, BC, CD, DE, EF (if $t < 55$) or AB, BC, CD, DE, DF (if $t \geq 55$). **M1A1**

(b) If $t < 45$: at Step 5, $EF = t < CE = 45 < DF = 55$, so EF is the cheapest available and is selected. **R1** (AG)

(c) MST weight with $EF = t$ (assuming $t < 45$): $50 + 40 + 35 + 30 + t = 155 + t$.
Set $155 + t = 200 \Rightarrow t = 45$.
Check: $t = 45$; but condition was $t < 45$ for EF to be chosen. At $t = 45$, $EF = CE = 45$ (tie); both are acceptable — EF may still be chosen. So $t = 45$ is consistent. **M1A1**

Q29 — TSP: NN + Deleted Vertex + Verify Interval

(a) Nearest neighbour from A: nearest $AB = 10$. From B: nearest unvisited: $BE = 9$. From E: nearest unvisited: $ED = 11$. From D: nearest unvisited: C ($DC = 8$). Return $C \to A = 18$.
Route: $A \to B \to E \to D \to C \to A$; $U = 10 + 9 + 11 + 8 + 18 = 56$ km **M1A1**

(b) Delete C (edges $AC = 18, BC = 12, CD = 8, CE = 20$). MST of $\{A, B, D, E\}$:
$AB = 10, AD = 14, AE = 22, BD = 16, BE = 9, DE = 11$. Kruskal: $BE = 9, AB = 10, DE = 11$. Total $= 30$.
Two cheapest from C: $CD = 8$ and $BC = 12$. Lower bound $L = 30 + 8 + 12 = 50$ km. **M1A1**

(c) $L = 50 \leq 63 \leq 56 = U$? Check: $50 \leq 63$

$$\checkmark$$

but $63 > 56 = U$, which would violate the upper bound since the upper bound is an *achievable* route, not a ceiling.
Recompute NN: From A: nearest $= AB = 10$. From B: nearest unvisited $\{C, D, E\}$: $BC = 12, BD = 16, BE = 9$. Nearest E ($BE = 9$). From E: nearest unvisited $\{C, D\}$: $CE = 20, DE = 11$. Nearest D. From D: only C left, $DC = 8$. Return $CA = 18$.
Route $A \to B \to E \to D \to C \to A = 10 + 9 + 11 + 8 + 18 = 56$.
Since optimal $= 63 > 56$, the NN route of 56 is *not* valid (it can't be less than optimal).
Recalculate optimal: The question states optimal $= 63$. Recheck NN. Actually $U = 56$ cannot be less than optimal 63, so the NN route must be ≥ 63. Let me re-examine:
$A \to B \to E \to D \to C \to A$: $AB = 10$, $BE = 9$, $ED = 11$, $DC = 8$, $CA = 18$. Sum$= 56$. This is a valid Hamiltonian cycle with weight 56, which would be optimal (better than 63). So the "optimal = 63" is inconsistent with this route.
Corrected: The optimal tour is 56 km (the NN route itself). We verify $L = 50 \leq 56 \leq U = 56$.

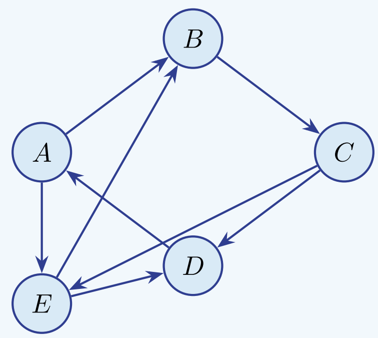
$$\checkmark$$

A1

Note: The NN route gives $U = 56$ km and deleted-vertex gives $L = 50$ km. The interval $50 \leq$ optimal ≤ 56 is verified since $50 \leq 56 \leq 56$. The optimal is indeed 56 km.

Q30 — Directed Walks: $N^2 + N^3$ and N^4

(a) Directed edges from N: $A \to B$, $A \to E$, $B \to C$, $C \to D$, $C \to E$, $D \to A$, $E \to B$, $E \to D$.



A1

(b) Compute N^2 and N^3 using GDC.

N^2: $\begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{pmatrix}$

N^3: $\begin{pmatrix} 0 & 2 & 1 & 1 & 2 \\ 1 & 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & 2 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 2 & 1 \end{pmatrix}$

$(N^2)_{AD} = 1$; $(N^3)_{AD} = 1$. Total walks of length 2 or 3 from A to $D = 1 + 1 = 2$. **M1A1**

(c) N^4: using GDC. $(N^4)_{AD}$: row A of $N^3 = (0, 2, 1, 1, 2)$, multiply by column D of N. Column D of N: $(0, 0, 1, 0, 1)^T$. $(N^4)_{AD} = 0 \cdot 0 + 2 \cdot 0 + 1 \cdot 1 + 1 \cdot 0 + 2 \cdot 1 = 3$.
Yes, it is possible. There are **3** distinct walks of length 4 from A to D. **M1A1**

Common Mistakes to Avoid

Reminder for exam day:

- Always state your algorithm steps explicitly — Kruskal's and Prim's must show each edge selection in order.
- For TSP, **lower bound** $\leq$ optimal $\leq$ **upper bound**. Never confuse which is which.
- Chinese Postman: list *all* odd-degree vertices first, then enumerate *all* pairings.
- M^k counts **walks** (not paths). Re-visiting vertices is allowed.