4.2 Decision Maths: Graphs, Trees and Minimum Spanning Trees: Test Pack

Calculator allowed · about 60 minutes

Total Marks

Name: _____ Date: ____ / ____ / _____

Answer ALL questions. Show all your working.

1 The graph G has vertex set {A, B, C, D, E} and edge set {AB, AC, BC, BD, CE, DE} (the graph is unweighted).

(a) State what is meant by a tree in graph theory. **(1)**

(b) State what is meant by a spanning tree of a connected graph. **(1)**

(c) For the graph G, state (i) the order of G, (ii) the size of G. **(2)**

(d) Determine, with justification, whether G is a tree. **(2)**

(e) State the number of edges in the complete graph K6. **(1)**

(Total for Question 1 is 7 marks)

---

2 A network has vertices P, Q, R, S and edges PQ=4, PR=6, PS=9, QR=3, QS=7, RS=5 (weights in km). A student claims that {PQ, PR, RS} is the minimum spanning tree for this network, with total weight 15 km.

(a) Explain why {PQ, PR, RS} is a valid spanning tree for this network, but state whether it is necessarily the minimum spanning tree. **(2)**

(b) Use Kruskal's algorithm to find the actual minimum spanning tree for this network, state its total weight, and hence confirm whether the tree given above is the minimum spanning tree. **(4)**

(Total for Question 2 is 6 marks)

© Revision Library | revisionlibrary.co.uk

3 The graph G has vertices A, B, C, D, E, F, G and edges AB, AC, AD, AG, BC, CD, CF, CG, DE, EF.

(a) Find the degree of each vertex, and verify the handshaking lemma for G. **(3)**

(b) Classify G as Eulerian, semi-Eulerian or neither, giving a reason. **(2)**

(c) The complete graph K_9 has 9 vertices, each joined to every other. Find its number of edges, and state whether it is Eulerian. **(3)**

(Total for Question 3 is 8 marks)

---

4 The graph G has vertices A, B, C, D, E, F, G and edges AB, AC, AG, BD, CE, CG, DE, DG, EF, EG, FG.

(a) Find the degree of each vertex, and verify the handshaking lemma for G. **(3)**

(b) Classify G as Eulerian, semi-Eulerian or neither, giving a reason. **(2)**

(c) The complete graph K_5 has 5 vertices, each joined to every other. Find its number of edges, and state whether it is Eulerian. **(3)**

(Total for Question 4 is 8 marks)

© Revision Library | revisionlibrary.co.uk

5 Six towns P, Q, R, S, T, U are to be linked by a new cable network. The possible direct routes and their lengths, in km, are: PQ=6, PR=9, PS=13, QR=4, QT=8, RS=5, RT=7, ST=6, SU=11, TU=3. Prim's algorithm, starting at P, is to be used to find a minimum spanning tree for this network.

(a) State how Prim's algorithm builds up a minimum spanning tree from a given starting vertex. **(2)**

(b) Apply Prim's algorithm to the network, starting at vertex P. Show the order in which vertices are added to the tree and the edge chosen at each step. **(6)**

(c) State the total weight of the minimum spanning tree found in part (b). **(1)**

(d) State two advantages of using Prim's algorithm, rather than Kruskal's algorithm, when working by hand with a large network given as a distance matrix. **(2)**

(Total for Question 5 is 11 marks)

---

6 Discuss the similarities and differences between Kruskal's algorithm and Prim's algorithm for finding a minimum spanning tree of a connected network. Your answer should refer to how each algorithm selects edges, how each avoids forming a cycle, and which algorithm is generally more efficient to apply by hand to a large, dense network given as a distance matrix.

(Total for Question 6 is 7 marks)

© Revision Library | revisionlibrary.co.uk

Mark scheme · 4.2 Decision Maths: Graphs, Trees and Minimum Spanning Trees: Test Pack

Question 1

(a) a connected graph containing no cycles (oe), e.g. a connected graph with exactly one path between every pair of vertices **B1**

(a) Answer: A tree is a connected graph that contains no cycles.

(b) a subgraph that includes every vertex of the original graph and is itself a tree (connected, no cycles) (oe) **B1**

(b) Answer: A spanning tree of a connected graph is a subgraph containing every vertex of the graph, which is connected and contains no cycles.

(c) order = 5 cao **B1**

(c) size = 6 cao **B1**

(c) Answer: Order = 5, size = 6.

(d) correct conclusion: G is not a tree **B1**

(d) valid justification, e.g. G contains the cycle A-B-C-A (oe), or: a tree on 5 vertices has exactly 4 edges but G has 6 **B1**

(d) Answer: G is not a tree.

(e) 15 cao **B1**

(e) Answer: 15

Question 2

(a) valid, since {PQ, PR, RS} has 3 (= 4 - 1) edges connecting all 4 vertices with no cycle, so it is a valid spanning tree **B1**

(a) not necessarily the minimum; a spanning tree is not automatically the minimum-weight spanning tree (oe) **B1**

(a) Answer: It is a valid spanning tree, but not necessarily the minimum one.

(b) edges sorted into ascending order: QR(3), PQ(4), RS(5), PR(6), QS(7), PS(9) **M1**

(b) QR, PQ, RS correctly accepted, forming a complete spanning tree; PR, QS, PS correctly rejected or not required **M1**

(b) total weight = 12 (km) cao **A1**

(b) since 12 < 15, the tree {PQ, PR, RS} is confirmed not to be the minimum spanning tree (oe) **A1**

(b) Answer: Minimum spanning tree = {QR, PQ, RS}, weight 12 km; the student's tree (weight 15) is not minimum.

Question 3

(a) counts the edges at each vertex **M1**

(a) degrees: A 4, B 2, C 5, D 3, E 2, F 2, G 2 **A1**

(a) sum of degrees = 20 = 2 x 10 edges **A1**

(a) Answer: A 4, B 2, C 5, D 3, E 2, F 2, G 2; sum 20

(b) counts the vertices of odd degree: C, D **M1**

(b) semi-Eulerian (has an Eulerian trail but no cycle) **A1**

(b) Answer: semi-Eulerian (has an Eulerian trail but no cycle)

(c) 9(9 - 1)/2 **M1**

(c) 36 edges **A1**

(c) each vertex has degree 8, which is even, so K_9 is Eulerian **B1**

(c) Answer: 36 edges; Eulerian

Question 4

(a) counts the edges at each vertex **M1**

(a) degrees: A 3, B 2, C 3, D 3, E 4, F 2, G 5 **A1**

(a) sum of degrees = 22 = 2 x 11 edges **A1**

(a) Answer: A 3, B 2, C 3, D 3, E 4, F 2, G 5; sum 22

(b) counts the vertices of odd degree: A, C, D, G **M1**

(b) neither Eulerian nor semi-Eulerian **A1**

(b) Answer: neither Eulerian nor semi-Eulerian

(c) 5(5 - 1)/2 **M1**

(c) 10 edges **A1**

(c) each vertex has degree 4, which is even, so K_5 is Eulerian **B1**

(c) Answer: 10 edges; Eulerian

Question 5

(a) the tree starts containing only the given starting vertex (oe) **B1**

(a) at each step, the least-weight edge joining a vertex already in the tree to a vertex not yet in the tree is added; this repeats until all vertices are included (oe) **B1**

(a) Answer: Start with only the given vertex; repeatedly add the cheapest edge from the tree to a vertex outside it, until all vertices are included.

(b) PQ=6 selected, Q added (rejecting PR=9, PS=13) **M1**

(b) QR=4 selected, R added **M1**

(b) RS=5 selected, S added **M1**

(b) ST=6 selected, T added (rejecting QT=8, RT=7) **M1**

(b) TU=3 selected, U added (rejecting SU=11) **A1**

(b) complete minimum spanning tree correctly stated as {PQ, QR, RS, ST, TU}, using 5 edges to connect all 6 vertices **A1**

(b) Answer: Order of addition: Q, R, S, T, U. Tree = {PQ, QR, RS, ST, TU}.

(c) 24 (km) ft their tree from part (b) **B1**

(c) Answer: 24 km

(d) Prim's algorithm always maintains a single growing tree, so there is no need to check explicitly whether an edge would create a cycle (oe) **B1**

(d) Kruskal's algorithm requires the whole edge list to be sorted in advance, which can be slow for a network with many edges; Prim's algorithm on a matrix only ever searches the rows/columns of vertices already in the tree (oe) **B1**

(d) Answer: Prim's algorithm needs no cycle-checking and avoids sorting the full edge list in advance.

© Revision Library | revisionlibrary.co.uk

Question 6

Level 0 No relevant comments made, or the response fails to address the question. **0**

Level 1 A basic description of at most one algorithm is given, with little or no valid comparison between the two. **1-2**

Level 2 Both algorithms are described with reasonable accuracy, and at least one valid comparison point is made (for example relating to cycle-checking or efficiency), but the discussion is not fully developed or contains minor inaccuracies. **3-5**

Level 3 Both algorithms are described accurately and at least three distinct, well-explained comparison points are made (for example cycle-checking, efficiency on dense versus sparse networks, and uniqueness of the resulting tree), with correct terminology used throughout. **6-7**

Indicative content:

- Both algorithms produce a spanning tree connecting all vertices with the minimum total weight, using exactly n - 1 edges.
- Kruskal's algorithm considers edges across the whole network in ascending order of weight, adding an edge unless it would form a cycle with edges already chosen; the structure being built can consist of several disconnected sub-trees before they eventually merge.
- Prim's algorithm always grows a single connected tree from a chosen starting vertex, at each stage adding the least-weight edge joining a vertex already in the tree to a vertex not yet in the tree.
- Because Prim's algorithm only ever has one connected component, there is no need to explicitly check whether adding an edge would create a cycle; Kruskal's algorithm does require this check whenever an edge under consideration could join two vertices already linked by other selected edges.
- Prim's algorithm applied to a distance matrix is generally more efficient to use by hand for a large, dense network, since at each stage only the rows and columns of vertices already in the tree need to be searched for the next minimum entry, with unwanted columns crossed out.
- Kruskal's algorithm requires the entire edge list to be sorted first, which can be time-consuming for a dense network with many edges, though it can be more straightforward to apply when the network is sparse (relatively few edges compared with vertices).
- Both algorithms are greedy algorithms (each makes the locally optimal choice at each stage) and both are guaranteed to produce a spanning tree of minimum total weight; the specific tree produced may not be unique when there are equal edge weights, but the minimum total weight itself is always the same.

© Revision Library | revisionlibrary.co.uk