NILE TUITION ACADEMY
Excellence in Further Maths Tuition
niletuitionacademy.co.uk

A Level Further Maths

Topic 5: Discrete (Optional)

AQA Further Maths (7367) | **Mark Scheme** | **100 marks**

Instructions

- Answer **all** questions. You must show sufficient working - answers with no working may score no marks.
- You may use a calculator unless a question states otherwise.
- Give exact answers where required, otherwise to an appropriate degree of accuracy.
- There are 10 questions in this paper. The total mark is **100**.

1. A simple connected graph G has 6 vertices and 9 edges.

(a) State the sum of the degrees (orders) of the vertices of G, giving a reason. **(2)**

✓ By the handshaking lemma the sum of degrees = 2 × (number of edges) (1)
✓ = 2 × 9 = 18 (1)

(b) Explain what is meant by saying that G is connected. **(1)**

✓ There is a path (route along edges) between every pair of vertices of G (1)

(c) The graph G is found to contain exactly one cycle removed from being a tree. State how many edges a spanning tree of G has, and how many edges would have to be deleted to obtain such a tree. **(2)**

✓ A spanning tree of a graph on 6 vertices has 6 - 1 = 5 edges (1)
✓ Edges to delete = 9 - 5 = 4 (1)

(d) Explain what is meant by a bipartite graph, and state the meaning of the complete bipartite graph $K_{3,3}$. **(3)**

✓ A bipartite graph has vertices in two disjoint sets with every edge joining a vertex of one set to a vertex of the other (1)
✓ $K_{3,3}$ has two sets of 3 vertices (1)
✓ with every vertex in one set joined to every vertex in the other (giving 9 edges) (1)

2. A network has five vertices A, B, C, D, E. The edge weights are: AB = 6, AC = 3, BC = 5, BD = 8, CD = 7, CE = 9, DE = 4.

(a) List the edges in order of increasing weight. **(1)**

✓ AC(3), DE(4), BC(5), AB(6), CD(7), BD(8), CE(9) (1)

(b) Apply Kruskal's algorithm to find a minimum spanning tree, stating clearly the order in which edges are chosen or rejected. **(4)**

✓ Choose AC(3); choose DE(4); choose BC(5) (1)
✓ Reject AB(6) (would form cycle A-B-C-A) (1)
✓ Choose CD(7), which connects {A,B,C} to {D,E} (1)
✓ Four edges now chosen connecting all five vertices, so stop (1)

(c) State the edges of the minimum spanning tree and its total weight. **(2)**

✓ Tree edges: AC, DE, BC, CD (1)
✓ Total weight = 3 + 4 + 5 + 7 = 19 (1)

(d) Explain why a minimum spanning tree of this network must contain exactly four edges. **(2)**

✓ A spanning tree on n vertices has n - 1 edges (1)
✓ Here n = 5, so the tree has 5 - 1 = 4 edges (1)

3. The distances, in km, between five towns P, Q, R, S, T are given in a network. The relevant edge weights are: PQ = 10, PR = 12, PS = 16, QR = 7, QS = 14, QT = 13, RS = 9, RT = 11, ST = 8.

(a) Starting from P, apply Prim's algorithm, listing the order in which the vertices are added to the tree. **(4)**

✓ Start P; nearest is Q (PQ = 10), add Q (1)
✓ From {P,Q} nearest new vertex is R (QR = 7), add R (1)
✓ From {P,Q,R} nearest is S (RS = 9), add S (1)
✓ From {P,Q,R,S} nearest is T (ST = 8), add T; order P, Q, R, S, T (1)

(b) State the edges used in the minimum spanning tree and its total length. **(2)**

✓ Edges: PQ, QR, RS, ST (1)
✓ Total length = 10 + 7 + 9 + 8 = 34 km (1)

(c) State one practical difference between the way Prim's algorithm and Kruskal's algorithm build the tree. **(2)**

✓ Prim grows a single connected tree from a starting vertex, always adding the nearest vertex not yet in the tree (1)
✓ Kruskal considers all edges in order of weight and may build several separate fragments that later join, rejecting any edge that forms a cycle (1)

(d) State one advantage of Prim's algorithm over Kruskal's when the network is given as a distance matrix. **(1)**

✓ Prim's algorithm works directly from the matrix without first having to sort all the edges by weight (1)

4. A network of one-way road sections connects vertices A, B, C, D, E. The arc weights (travel times in minutes) are: AB = 4, AC = 2, CB = 1, BD = 5, CD = 8, CE = 10, DE = 2.

(a) Use Dijkstra's algorithm to find the shortest time from A to each other vertex. Show the final (permanent) label at each vertex. **(5)**

✓ A: 0 (permanent first) (1)
✓ C: 2 (via A); then B: min(4 via A, 2 + 1 via C) = 3 (1)
✓ D: 3 + 5 = 8 (via B) (1)
✓ E: min(2 + 10 via C, 8 + 2 via D) = 10 (via D) (1)
✓ Final labels A = 0, C = 2, B = 3, D = 8, E = 10 (1)

(b) State the shortest route from A to E and its length. **(2)**

✓ Shortest route A - C - B - D - E (1)
✓ Length = 2 + 1 + 5 + 2 = 10 minutes (1)

(c) Explain how, once the algorithm is complete, the shortest route to a vertex is traced back. **(2)**

✓ Start at the destination and work backwards (1)
✓ At each vertex choose the previous vertex whose permanent label plus the connecting arc equals this vertex's label (1)

(d) State one assumption about the arc weights needed for Dijkstra's algorithm to be valid. **(1)**

✓ All arc weights must be non-negative (1)

5. A network of paths must be inspected by a worker who starts and finishes at the same point and walks along every path. The four odd vertices of the network are A, B, C, D, with the shortest path distances between them being AB = 8, AC = 5, AD = 9, BC = 7, BD = 6, CD = 4. The total length of all the paths is 120 m.

(a) Explain why the route inspection (Chinese postman) problem requires the odd vertices to be paired up. **(2)**

✓ A closed route using every edge exists without repetition only if every vertex has even degree (1)
✓ The odd vertices must be paired and the paths between each pair repeated to make all degrees even (1)

(b) List the three ways of pairing the four odd vertices and find the total of the shortest distances for each pairing. **(3)**

✓ AB + CD = 8 + 4 = 12 (1)
✓ AC + BD = 5 + 6 = 11 (1)
✓ AD + BC = 9 + 7 = 16 (1)

(c) State the minimum total distance the worker must walk to inspect every path and return to the start. **(2)**

✓ Choose the smallest repeat total: AC + BD = 11 (1)
✓ Minimum distance = 120 + 11 = 131 m (1)

(d) Explain the difference between the route inspection problem and the travelling salesperson problem. **(2)**

✓ In route inspection every edge (path) must be traversed, at least once, returning to the start (1)
✓ In the travelling salesperson problem every vertex must be visited (once) and a minimum-weight Hamiltonian cycle is sought, not every edge (1)

6. A salesperson must visit five towns and return to the start. A lower bound for the optimal tour is found by deleting a vertex and adding to the minimum spanning tree of the rest the two shortest arcs from the deleted vertex. The minimum spanning tree of the remaining four towns has weight 40, and the two shortest arcs from the deleted town have lengths 9 and 11.

(a) Calculate this lower bound for the length of the optimal tour. **(2)**

✓ Lower bound = MST weight + two shortest arcs = 40 + 9 + 11 (1)
✓ = 60 (1)

(b) Using a nearest neighbour algorithm from the start town a tour of length 72 is found. Write down an inequality satisfied by the length L of the optimal tour. **(2)**

✓ A nearest neighbour tour gives an upper bound, the lower bound is 60 (1)
✓ 60 <= L <= 72 (1)

(c) Explain why the nearest neighbour algorithm does not always produce the optimal tour. **(2)**

✓ At each step it greedily takes the nearest unvisited town (1)
✓ An early short arc can force long arcs later, so the overall tour may not be the shortest (it is a heuristic, not exact) (1)

(d) Explain why the travelling salesperson problem is hard to solve exactly for large networks, and state what is meant by a heuristic algorithm in this context. **(4)**

✓ The number of distinct tours grows extremely rapidly (roughly factorially) with the number of towns (1)
✓ so checking every tour becomes computationally infeasible for large n (1)
✓ A heuristic algorithm produces a good (often near-optimal) solution (1)
✓ quickly, but without guaranteeing that it is the exact optimal tour (1)

7. This question is about standard algorithms applied to the list of numbers 8, 3, 5, 9, 2, 6.

(a) Perform one pass of a bubble sort (sorting into increasing order), writing down the list after the pass. **(3)**

✓ Compare and swap adjacent pairs from the left: 3, 8, 5, 9, 2, 6 then 3, 5, 8, 9, 2, 6 (1)
✓ 8 and 9 not swapped; then 9 and 2 swap: 3, 5, 8, 2, 9, 6 (1)
✓ 9 and 6 swap to give 3, 5, 8, 2, 6, 9 (largest value 9 now at the end) (1)

(b) State how many comparisons are made in one complete pass of a bubble sort on this list of six numbers. **(1)**

✓ 6 - 1 = 5 comparisons (1)

(c) Boxes of capacity 10 are to be filled with items of sizes 8, 3, 5, 9, 2, 6 (in this order). Apply the first-fit algorithm and state how many boxes are used. **(3)**

✓ Take the items in the given order and place each in the first box with room. 8 -> Box 1; 3 -> Box 2 (8 + 3 = 11 > 10, will not fit in Box 1) (1)
✓ 5 -> Box 2 (3 + 5 = 8); 9 -> Box 3 (will not fit in Box 1 or Box 2) (1)
✓ 2 -> Box 1 (8 + 2 = 10); 6 -> Box 4 (will not fit in Box 1 full, Box 2 = 8 + 6 = 14, or Box 3 = 9 + 6 = 15). Boxes: Box 1: 8, 2; Box 2: 3, 5; Box 3: 9; Box 4: 6. Total = 4 boxes (1)

(d) Apply the first-fit decreasing algorithm to the same items and capacity, and state whether it uses fewer boxes. **(3)**

✓ Sort into decreasing order: 9, 8, 6, 5, 3, 2. Place each in the first box with room. 9 -> Box 1; 8 -> Box 2; 6 -> Box 3 (1)
✓ 5 -> Box 4 (will not fit: 9 + 5, 8 + 5, 6 + 5 all exceed 10); 3 -> Box 3 (6 + 3 = 9); 2 -> Box 2 (8 + 2 = 10). Boxes: Box 1: 9; Box 2: 8, 2; Box 3: 6, 3; Box 4: 5. Total = 4 boxes (1)
✓ First-fit decreasing also uses 4 boxes here, so it does not use fewer (the lower bound is ceil(33/10) = 4 boxes) (1)

8. A project consists of activities A to F with the following durations (days) and immediate predecessors: A(3, none), B(4, none), C(2, A), D(5, A and B), E(6, C), F(4, D).

(a) Find the early event times by carrying out a forward pass through the network. **(3)**

✓ Start = 0; after A: 3, after B: 4 (1)
✓ C can start at 3 (ends 5); D starts after A and B, at max(3, 4) = 4 (ends 9) (1)
✓ E starts at 5 (ends 11); F starts at 9 (ends 13); project early finish = 13 (1)

(b) Find the late event times by carrying out a backward pass, and hence state the minimum project completion time. **(3)**

✓ Project finish = 13; late finish of E = 13 (start 7), late finish of F = 13 (start 9) (1)
✓ Late times work back: D late finish 9 (start 4), C late finish 7 (start 5), A late finish min(start of C, start of D) and B late finish 4 (1)
✓ Minimum project completion time = 13 days (1)

(c) List the critical activities and state the critical path. **(2)**

✓ Critical activities are those with zero float: B, D, F (B start 0, D start 4, F start 9, all on the longest path) (1)
✓ Critical path: B - D - F (length 4 + 5 + 4 = 13) (1)

(d) Calculate the total float on activity C. **(2)**

✓ Total float = latest finish - earliest start - duration = 7 - 3 - 2 (1)
✓ = 2 days (1)

9. A manufacturer makes x units of product P and y units of product Q. The profit, in pounds, is given by 5x + 4y, subject to 2x + y <= 20, x + 3y <= 30, and x, y >= 0.

(a) Write the linear programming problem as a maximisation in standard form, introducing slack variables s and t. **(3)**

✓ Maximise P = 5x + 4y (1)
✓ subject to 2x + y + s = 20 and x + 3y + t = 30 (1)
✓ with x, y, s, t >= 0 (1)

(b) Write down the initial simplex tableau for this problem. **(2)**

✓ Constraint rows: (2, 1, 1, 0 | 20) and (1, 3, 0, 1 | 30) (1)
✓ Objective row: P - 5x - 4y = 0, i.e. (-5, -4, 0, 0 | 0) (1)

(c) Identify the pivot column and pivot element for the first iteration, explaining your choice. **(3)**

✓ Pivot column is x: it has the most negative entry (-5) in the objective row (1)
✓ Ratio test: 20/2 = 10 and 30/1 = 30; the smaller ratio is 10 (1)
✓ So the pivot is in the first row, pivot element = 2 (entry in the x-column, first row) (1)

(d) By solving the problem graphically or otherwise, find the values of x and y that maximise the profit and state the maximum profit. **(4)**

✓ The vertex at the intersection of 2x + y = 20 and x + 3y = 30: from the first y = 20 - 2x (1)
✓ x + 3(20 - 2x) = 30 gives x + 60 - 6x = 30, so -5x = -30, x = 6, y = 8 (1)
✓ Check other vertices: (10, 0) gives 50, (0, 10) gives 40, (6, 8) gives 5(6) + 4(8) = 62 (1)
✓ Maximum profit = 62 pounds at x = 6, y = 8 (1)

10. A two-player zero-sum game has the pay-off matrix for player A (row player), with player B as the column player: row strategy A1 gives (3, 1) against B's columns, and row strategy A2 gives (0, 4).

(a) Determine the row maximin and the column minimax, and hence explain whether the game has a stable solution (saddle point). **(4)**

✓ Row minima: A1 gives min(3, 1) = 1, A2 gives min(0, 4) = 0; row maximin = 1 (1)
✓ Column maxima: column 1 gives max(3, 0) = 3, column 2 gives max(1, 4) = 4; column minimax = 3 (1)
✓ Maximin (1) does not equal minimax (3) (1)
✓ So there is no saddle point; the game is not stable and requires a mixed strategy (1)

(b) Player A plays A1 with probability p and A2 with probability 1 - p. Write expressions for A's expected pay-off against each of B's pure strategies. **(2)**

✓ Against B column 1: E_1 = 3p + 0(1 - p) = 3p (1)
✓ Against B column 2: E_2 = 1p + 4(1 - p) = 4 - 3p (1)

(c) Find the value of p for A's optimal mixed strategy and the value of the game. **(3)**

✓ Equate the expected pay-offs: 3p = 4 - 3p (1)
✓ 6p = 4, so p = 2/3 (1)
✓ Value of game = 3p = 3(2/3) = 2 (1)

(d) A separate transport network has a source S and sink T. A cut separates S from T and consists of the arcs with capacities 7, 5 and 4. State the capacity of this cut and explain what the maximum flow - minimum cut theorem tells you about the maximum flow. **(4)**

✓ Capacity of the cut = sum of the capacities of its arcs = 7 + 5 + 4 = 16 (1)
✓ The maximum flow from S to T cannot exceed the capacity of any cut (1)
✓ The maximum flow - minimum cut theorem states the maximum flow equals the capacity of the minimum cut (1)
✓ so the maximum flow is at most 16, and equals 16 if this is the minimum cut (1)