4.5 Decision: Algorithms and Graph Theory Depth: Test Pack

Total Marks

EDEXCEL 9FM0 · Calculator allowed · about 60 minutes

Name: _____ Date: ____ / ____ / _____

Answer ALL questions. Show all your working.

1 The following list of six numbers is to be sorted into ascending order:
8, 3, 10, 5, 1, 7

(a) Perform a bubble sort on the list, comparing and swapping adjacent items from left to right. Show the list in full after each pass, and continue until you have completed one full pass in which no swaps are made. **(4)**

(b) State the number of comparisons made during the first pass. **(1)**

(Total for Question 1 is 5 marks)

---

2 This question concerns the efficiency of sorting algorithms as the size of a list, n, increases.

(a) State the order of the bubble sort algorithm, in terms of the number of items n, for the worst-case number of comparisons required. **(1)**

(b) Given that, in the worst case, a bubble sort on a list of n items requires exactly $n(n - 1)/2$ comparisons, calculate this number of comparisons for a list of $n = 60$ items. **(2)**

(c) A quicksort algorithm has worst-case order $O(n^2)$, the same as bubble sort, but average-case order $O(n \log n)$. Explain why quicksort is generally preferred over bubble sort in practice, despite sharing the same worst-case order. **(2)**

(Total for Question 2 is 5 marks)

© Revision Library | revisionlibrary.co.uk

3 A haulage firm must load crates of the following weights, in kg, onto lorries that can each carry a maximum of 50 kg:
22, 28, 19, 31, 12, 38

(a) Using a full-bin method, find combinations of crates that exactly fill a 50 kg lorry, and show that all six crates can be loaded using exactly three lorries with no wasted capacity. **(3)**

(b) Calculate a lower bound for the number of lorries required, and state, with a reason, whether the packing found in part (a) is optimal. **(2)**

(Total for Question 3 is 5 marks)

---

4 This question concerns further properties of graphs and networks.

(a) A connected planar graph has 9 vertices and 14 edges. Use Euler's formula for planar graphs, $v - e + f = 2$, to find the number of faces of the graph (including the unbounded outer face). **(2)**

(b) State what is meant by a bipartite graph, and calculate the number of edges in the complete bipartite graph K(4,5), which has one set of 4 vertices and another set of 5 vertices, with every vertex in one set joined to every vertex in the other set, and no edges within either set. **(2)**

(c) State what is meant by a Hamiltonian cycle in a graph, and explain how this differs from an Eulerian circuit. **(2)**

(d) Explain why every tree must be a bipartite graph. **(2)**

(Total for Question 4 is 8 marks)

© Revision Library | revisionlibrary.co.uk

5 A logistics manager is planning work on the same road network connecting a depot and several towns, and has three separate problems to solve:
Problem 1: connect every town to the depot using cable of minimum total length (the towns do not need to be able to reach each other quickly afterwards, only to be linked into a single network).
Problem 2: find the quickest possible route for a single van to drive from the depot to one specific town.
Problem 3: find the shortest possible route for a gritting lorry that must drive along every road in the network at least once, starting and finishing at the depot.

Discuss which type of algorithm, a minimum spanning tree algorithm (Kruskal's or Prim's), Dijkstra's algorithm, or the route inspection algorithm, should be used to solve each problem. Your answer should explain what feature of each problem identifies the correct technique, and explain why using an incorrect technique would fail to answer the question actually being asked.

(Total for Question 5 is 6 marks)

---

6 The items with the following sizes are to be packed into bins of capacity 50: 11, 19, 25, 42, 18, 11, 42, 21, 5.

(a) Find a lower bound for the number of bins needed. **(2)**

(b) Use the first-fit algorithm, taking the items in the order given, to pack the items into bins. State the contents of each bin. **(3)**

(c) Use the first-fit decreasing algorithm to pack the items. State the number of bins used and the contents of each bin. **(3)**

(Total for Question 6 is 8 marks)

---

7 A network of road junctions, A to G, used by a fleet of breakdown-recovery vans, has direct roads with journey times, in minutes: AB = 4, AC = 7, BC = 2, BD = 6, CD = 3, CE = 8, DE = 1, DF = 5, EF = 4, EG = 7, FG = 2

(a) Use Dijkstra's algorithm, starting at A, to find the shortest time from junction A to every other junction. Show clearly the working values considered, and the order in which each junction receives its final (permanent) label. **(8)**

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

(Total for Question 7 is 10 marks)

© Revision Library | revisionlibrary.co.uk

Mark scheme · 4.5 Decision: Algorithms and Graph Theory Depth: Test Pack

Question 1

(a) pass 1 correct: 3, 8, 5, 1, 7, 10 **M1**

(a) pass 2 correct: 3, 5, 1, 7, 8, 10 **A1**

(a) passes 3 and 4 both correct: 3, 1, 5, 7, 8, 10 then 1, 3, 5, 7, 8, 10 **A1**

(a) a further (5th) pass shown with no swaps, confirming the list is sorted **B1**

(a) Answer: Pass 1: 3, 8, 5, 1, 7, 10. Pass 2: 3, 5, 1, 7, 8, 10. Pass 3: 3, 1, 5, 7, 8, 10. Pass 4: 1, 3, 5, 7, 8, 10. Pass 5 (check): 1, 3, 5, 7, 8, 10 (no swaps).

(b) 5 comparisons cao **B1**

(b) Answer: 5 comparisons.

Question 2

(a) $O(n^2)$ **B1**

(a) Answer: $O(n^2)$.

(b) substitutes n = 60 into n(n - 1)/2 **M1**

(b) 1770 cao **A1**

(b) Answer: 1770 comparisons.

(c) quicksort's average-case order, O(n log n), grows far more slowly than bubble sort's typical performance, which is close to $O(n^2)$ for almost all inputs (oe) **B1**

(c) in practice, most real lists do not trigger quicksort's worst case, so it usually runs much closer to its efficient average-case behaviour, making it faster than bubble sort for large lists in the great majority of cases (oe) **B1**

(c) Answer: Quicksort's average-case order O(n log n) is far better than bubble sort's near-$O(n^2)$ typical performance; since quicksort's worst case is rare in practice, it is usually much faster for large lists.

Question 3

(a) identifies at least one pair of crates summing to exactly 50 kg, e.g. 22 + 28 = 50 **M1**

(a) all three valid pairs identified: (22, 28), (19, 31) and (12, 38), each summing to exactly 50 **A1**

(a) states three lorries, each exactly full: Lorry 1 {22, 28}, Lorry 2 {19, 31}, Lorry 3 {12, 38} **A1**

(a) Answer: Lorry 1: 22, 28 (50 kg). Lorry 2: 19, 31 (50 kg). Lorry 3: 12, 38 (50 kg). 3 lorries, all exactly full.

(b) total weight = 150 kg, lower bound = ceiling(150/50) = 3 **B1**

(b) yes, optimal, since the packing in part (a) achieves this lower bound of 3 lorries with no wasted capacity (oe) **B1**

(b) Answer: Lower bound = 3 lorries; the packing in part (a) achieves this, so it is optimal.

Question 4

(a) substitutes v = 9 and e = 14 into v - e + f = 2 **M1**

(a) f = 7 cao **A1**

(a) Answer: 7 faces.

(b) a bipartite graph is one whose vertices can be split into two disjoint sets so that every edge joins a vertex in one set to a vertex in the other set, with no edges within either set (oe) **B1**

(b) number of edges in K(4,5) = 4*5 = 20 cao **B1**

(b) Answer: A bipartite graph splits its vertices into two sets with edges only between the sets, never within a set; K(4,5) has 4*5 = 20 edges.

(c) a Hamiltonian cycle is a cycle that visits every vertex of the graph exactly once, before returning to its starting vertex (oe) **B1**

(c) an Eulerian circuit instead requires every edge (not vertex) to be used exactly once, returning to the start; a graph may possess one of these properties without the other (oe) **B1**

(c) Answer: A Hamiltonian cycle visits every vertex exactly once; an Eulerian circuit instead uses every edge exactly once. A graph can have either property without the other.

(d) a tree contains no cycles at all, by definition, so in particular it contains no cycle of odd length (oe) **B1**

(d) a graph is bipartite if and only if it contains no odd-length cycle; since a tree has no cycles whatsoever, this condition is automatically satisfied, so every tree is bipartite (oe) **B1**

(d) Answer: A tree has no cycles at all, so it certainly has no odd cycles; since a graph is bipartite exactly when it has no odd cycle, every tree must be bipartite.

© Revision Library | revisionlibrary.co.uk

Question 5

Level 0 No relevant technique correctly matched to any of the three problems, or the response does not address the question.	**0**
Level 1 At least one problem is matched to a broadly appropriate technique, but with little or no valid justification, and no discussion of why other techniques would be unsuitable.	**1-2**
Level 2 At least two of the three problems are correctly matched to their appropriate technique, with some valid justification referring to what each problem requires; explanation of why an incorrect technique would fail is limited or only partially developed.	**3-4**
Level 3 All three problems are correctly matched to their appropriate technique (minimum spanning tree algorithm for Problem 1, Dijkstra's algorithm for Problem 2, route inspection algorithm for Problem 3), with clear justification referring to the specific requirement of each problem, and a clear explanation of why using an incorrect technique for a given problem would fail to answer the question actually being asked.	**5-6**

Indicative content:

- Problem 1 only requires every town to be linked into a single connected network using the least total length of cable, with no requirement to travel quickly between any two particular towns afterwards; this is a minimum spanning tree problem, solved using Kruskal's algorithm or Prim's algorithm.
- Problem 2 requires the shortest route between two specific vertices only (the depot and one named town); this is exactly what Dijkstra's algorithm finds.
- Problem 3 requires a route that uses every road in the network at least once and returns to its starting point; this is the route inspection (Chinese postman) problem.
- Using Dijkstra's algorithm for Problem 1 would only ever find a path between two particular vertices, not a structure connecting every town together, so it cannot solve Problem 1.
- Using a minimum spanning tree for Problem 2 would not generally give the shortest route between the depot and a specific town, since a spanning tree minimises the total length of the whole network, not the distance between any one particular pair of vertices; the route between two towns within a minimum spanning tree can be much longer than the true shortest path between them.
- Using a minimum spanning tree, or Dijkstra's algorithm, for Problem 3 would not guarantee that every road is used: a spanning tree deliberately omits some edges (it has only n - 1 edges for n vertices), and Dijkstra's algorithm only ever finds a route between two vertices, not a route covering every edge.
- The key skill is recognising the single feature that identifies each type of problem: connect everything as cheaply as possible (minimum spanning tree); find the shortest distance between two specific vertices (Dijkstra's algorithm); or use every edge at least once (route inspection algorithm).

Question 6

(a) total size = 194	**M1**
(a) 194/50 = 3.88, so at least 4 bins	**A1**
(a) Answer: 4	
(b) places each item in the first bin with enough room, in the order given	**M1**
(b) first bins correct	**A1**
(b) bins: (11, 19, 18); (25, 11, 5); (42); (42); (21)	**A1**
(b) Answer: (11, 19, 18); (25, 11, 5); (42); (42); (21)	
(c) sorts the items into descending order	**M1**
(c) applies first-fit to the sorted list	**M1**
(c) 5 bins: (42, 5); (42); (25, 21); (19, 18, 11); (11)	**A1**
(c) Answer: 5 bins: (42, 5); (42); (25, 21); (19, 18, 11); (11)	

Question 7

(a) A permanently labelled with 0; initial working values B = 4 (via A), C = 7 (via A)	**M1**
(a) B permanently labelled second, with value 4; C updated to 6 (via B, since 4 + 2 = 6 < 7)	**A1**
(a) C permanently labelled third, with value 6; D = 9 (via C), E = 14 (via C)	**A1**
(a) D permanently labelled fourth, with value 9; E updated to 10 (via D, since 9 + 1 = 10 < 14), F = 14 (via D)	**A1**
(a) E permanently labelled fifth, with value 10; G = 17 (via E); notes F's candidate value via E (10 + 4 = 14) does not improve on F's existing value of 14 (via D), so F is unchanged	**A1**
(a) F permanently labelled sixth, with value 14 (via D)	**A1**
(a) G updated to 16 (via F, since 14 + 2 = 16 < 17)	**A1**
(a) G permanently labelled seventh (last), with value 16; full order of labelling correctly stated as A, B, C, D, E, F, G	**B1**
(a) Answer: Order of permanent labelling: A(0), B(4), C(6), D(9), E(10), F(14), G(16).	
(b) A-B-C-D-F-G (oe, traced back through predecessors from G); also accept A-B-C-D-E-F-G, which is also 16 minutes (F can be reached via D or via E)	**B1**
(b) 16 (minutes) cao	**B1**
(b) Answer: A-B-C-D-F-G, 16 minutes (A-B-C-D-E-F-G is also 16 minutes).	

© Revision Library | revisionlibrary.co.uk