

2.3 ALGORITHMS · 2.3.1(E)

Graph traversals — Mark scheme

30 marks · spec 2.3.1(e)

AO key: AO1 = knowledge · AO2 = application. Traversal orders verified (start P, neighbours alphabetical).

Q	ANSWER	AO	MARKS
1(a)	A systematic way of visiting every node in a graph (once). (1)	AO1	1
1(b)	Goes as deep as possible then backtracks at dead ends (1); uses a stack (or recursion) (1). (2)	AO1	2
1(c)	Visits nodes layer by layer (neighbours first) (1); uses a queue (1). (2)	AO1	2
1(d)	Breadth-first (BFS). (1)	AO1	1

Q	ANSWER	AO	MARKS
2(a)	DFS: P, Q, S, R, T. Fully correct (3); mostly correct (2); right method, ≥ 2 right (1). (3)	AO2	3
2(b)	BFS: P, Q, R, S, T. Fully correct (3); mostly correct (2); right method, ≥ 2 right (1). (3)	AO2	3
2(c)	A queue. (1)	AO1	1

Q	ANSWER	AO	MARKS
3(a)	1 mark each, max 2: finding a path · detecting cycles · solving mazes (backtracking) · topological sort. (2)	AO1	2
3(b)	1 mark each, max 2: shortest path in unweighted graph · web crawlers · exploring by distance from start. (2)	AO1	2
3(c)	It visits nodes in order of distance, so the target is first reached by the fewest edges. (1)	AO1	1

Q	ANSWER	AO	MARKS
4(a)	Breadth-first traversal (BFS). (1)	AO2	1
4(b)	BFS explores by distance / layer (1); so it reaches the destination by the fewest hops (shortest unweighted path) (1). (2)	AO2	2
4(c)	A queue. (1)	AO1	1
4(d)	DFS plunges deep down one path first (1); so it may reach the target by a long route before a shorter one, not guaranteeing fewest hops (1). (2)	AO2	2

Q	ANSWER	AO	MARKS
5(a)	DFS goes deep along one path before backtracking (1); BFS spreads out layer by layer from the start (1). (2)	AO1	2
5(b)	Post-order. (1)	AO1	1
5(c)	To avoid revisiting nodes (1); which would otherwise cause repeated work or an infinite loop around cycles (1). (2)	AO2	2
5(d)	It uses a lot of memory (must store a whole layer / large queue). (1)	AO1	1

Total for paper: 30 marks