2.3 ALGORITHMS · 2.3.1(F)

Dijkstra's algorithm & A*

Original practice questions · 30 marks · about 45 minutes · spec 2.3.1(f)

Instructions. Answer all questions. Marks are shown in brackets [ ]. Show working for the shortest-path trace.

1

Total: 6 marks

(a) State what Dijkstra's algorithm finds. _____ **[1]**

(b) State what kind of graph Dijkstra's algorithm needs. _____ **[1]**

(c) Describe how Dijkstra's algorithm chooses which node to settle next. **[2]**

(d) State what the algorithm records so the shortest path can be reconstructed. **[2]**

2

Total: 8 marks

```
Weighted edges: S–A 2, S–B 5, A–B 1, A–C 4, B–C 2, B–D 7, C–D 1 (start at S)
```

(a) Complete the table of the final shortest distance from S to each node. **[5]**

Node	S	A	B	C	D
Dist from S	0				

(b) State the shortest path from S to D, and its total length. _____ **[2]**

(c) State which node is settled first after S. _____ **[1]**

3 Total: 6 marks

(a) Explain how the A* algorithm improves on Dijkstra's when searching for a specific goal. **[2]**

(b) State the cost function A* uses to choose the next node. **[1]**

(c) State what g and h represent in that function. **[2]**

(d) State the condition under which A* is guaranteed to find the shortest path. **[1]**

4 Total: 4 marks

(a) State one difference between Dijkstra's algorithm and A*. **[2]**

(b) Explain why A* usually explores fewer nodes than Dijkstra's. **[2]**

5 Total: 6 marks

```
A sat-nav must find the quickest route between two places on a road network.
```

(a) Explain why A* is well suited to this task. **[2]**

(b) Suggest a sensible heuristic h for road routing. **[1]**

(c) State how A* behaves if the heuristic is zero everywhere. **[1]**

(d) Explain why the heuristic must not overestimate the distance to the goal. **[2]**

END OF QUESTIONS · END OF PAPER 2 SERIES