inkMaths

ANSWER BOOK

Dynamic programming

Decision Mathematics 2 · Year FM

6 questions · 23 marks

NAVY EDITION

inkphysics.co.uk/maths

SECTION A · Warm-up

A1 Any part of an optimal path is itself optimal between its own two ends. It allows the problem to be solved **backwards**: once the best value from every vertex onwards is known, the best value at the vertex before it is one comparison away, and no route already rejected needs reconsidering. B1 for the principle, B1 for working backwards. [2]

A2 Minimum total: min(6 + 11, 10 + 5) = min(17, 15) = **15**, through the second option. Minimax: min(max(6, 11), max(10, 5)) = min(11, 10) = **10**, also through the second, but chosen because its worst single stage is smaller rather than because its total is. M1 for adding along each option, A1 for 15, B1 for 10 under the minimax rule. [3]

SECTION B · Standard

B1 Work backwards, one row per stage, state, action and value. [6]
Stage 3 (state, action, value): C, CT, **6**; D, DT, **7**.
Stage 2: A, AC, 4 + 6 = 10*; A, AD, 9 + 7 = 16; so A is worth **10** through C. B, BC, 8 + 6 = 14; B, BD, 2 + 7 = 9*; so B is worth **9** through D.
Stage 1: S, SA, 5 + 10 = 15; S, SB, 3 + 9 = 12*; so S is worth **12** through B.
Reading the starred actions forwards gives the route **S,** B, D, T of length **12**. M1 for working backwards in stages, A1 for the final stage values, M1 for the middle stage, A1 for its two values, A1 for the value at S, A1 for the route. Checking all four routes gives 12, 15, 17 and 21, confirming it. Marks are for the table itself, so every option must appear, not only the winner.

B2 The two happen to agree here, which is exactly why the example is a poor test of the method. The choice was made by comparing 5 + 10 with 3 + 9, both of which use values found from the far end. Had the value at A been 5 rather than 10, S to A would have won at 10 despite the dearer first step. A greedy first move carries no guarantee about what follows it. B1 for the comparison using values from the far end, B1 for the altered example, B1 for a greedy first step giving no guarantee. [3]

B3 The **stage** says how far through the process you are: here there are three stages, one per arc travelled. The **state** says where you are within that stage. At stage 2 the states are A and B, and at stage 3 they are C and D. B1 for the stage, B1 for the state, B1 for the states named in this network. Every row of the table is one stage and state pair together with one action, and naming all three is part of the answer. [3]

SECTION C · Stretch

C1 This is a **maximin** problem, so the addition is replaced by taking the minimum along a route. [6]
Stage 3: C is worth 6 and D is worth 7, as before.
Stage 2: A gives max(min(4, 6), min(9, 7)) = max(4, 7) = **7** through D; B gives max(min(8, 6), min(2, 7)) = max(6, 2) = **6** through C.
Stage 1: S gives max(min(5, 7), min(3, 6)) = max(5, 3) = **5** through A.
The route is **S,** A, D, T with smallest stage **5**. B1 for replacing addition by the minimum along a route, M1 for the middle stage, A1 for its two values, M1 for the first stage, A1 for the value 5, A1 for the route. It is a different route from the shortest one, which shows that the two objectives genuinely pull apart: S, A, D, T is the longest route of the four by total length, at 21.