Exercise walk-through

10.3
Route inspection and the travelling salesman problem

Pearson Edexcel · International A-Level · Further Mathematics

The travelling salesman problem

- **Upper bound**: an actual route. Doubling the weight of the MST gives one; shortcuts improve it. The **nearest neighbour algorithm** also gives one: from the start, go to the nearest unvisited vertex each time, then return to the start.
- **Lower bound**: delete one vertex, find the MST of the rest, and add the two smallest edges from the deleted vertex.
- Nearest neighbour from A: A–C (2), C–B (3), B–D (5), D–A (7). Total 17, an upper bound.

The travelling salesman problem (2)

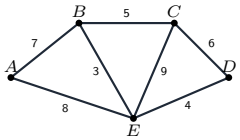
- The MST is AC, BC, BD, with total 10, so doubling gives the weaker upper bound 20.
- Lower bound, deleting A: the MST of B, C, D is BC + BD = 8, and the two smallest edges from A are AC + AB = 6. So the lower bound is 14.
- So $14 \leq$ optimal route ≤ 17. With only four vertices, we can check all three routes: A–B–C–D–A = 20, A–B–D–C–A = 17 and A–C–B–D–A = 17. The optimal route has length 17.

Key terms

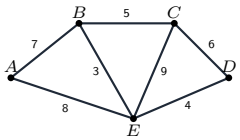
- **Minimum spanning tree**: a tree connecting all the vertices with the least total weight.
- **Dijkstra's algorithm**: an algorithm for the shortest path from one vertex to the others.
- **Route inspection problem**: finding the shortest closed route that uses every edge.

Exercises 2.1·2.2

2.1 A network has edges AB 5, BC 2, CD 4, DA 3, AC 6 with total 20. Find the odd nodes, the least duplication, and the length of the shortest closed route covering every edge. [4]

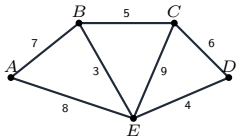


2.2 For a complete network on W, X, Y, Z with WX 3, WY 4, WZ 5, XY 2, XZ 6, YZ 7: find the nearest neighbour tour from W and its length. [3]



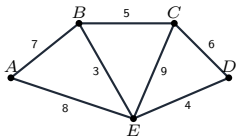
Exercises 2.1·2.2

2.1 A network has edges AB 5, BC 2, CD 4, DA 3, AC 6 with total 20. Find the odd nodes, the least duplication, and the length of the shortest closed route covering every edge. [4]



2.1 Degrees: A 3 (AB, AD, AC), B 2, C 3 (BC, CD, CA), D 2 – odd nodes A and C. The only pairing is AC (6): route length $20 + 6 = 26$.

2.2 For a complete network on W, X, Y, Z with WX 3, WY 4, WZ 5, XY 2, XZ 6, YZ 7: find the nearest neighbour tour from W and its length. [3]

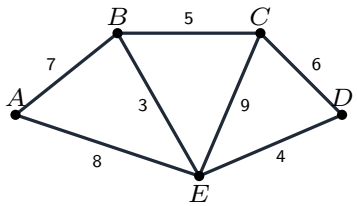


2.2 From W: nearest X (3); from X: nearest Y (2); from Y: nearest Z (7)? YZ = 7 vs YW = 4 but W is visited – from Y the unvisited are Z only: Y–Z (7); return Z–W (5). Tour W–X–Y–Z–W = $3 + 2 + 7 + 5 = 17$.

exercises.lol exercises.com.cn

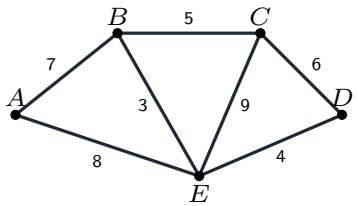
Exercise 2.3

2.3 The MST of a complete network (triangle inequality holding) weighs 12. State the upper bound obtained by doubling the MST, and explain in one sentence why shortcuts can only improve it. [2]



Exercise 2.3

2.3 The MST of a complete network (triangle inequality holding) weighs 12. State the upper bound obtained by doubling the MST, and explain in one sentence why shortcuts can only improve it. [2]

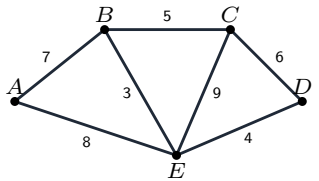


2.3 Doubled MST: 24. A closed walk along each MST edge twice visits every vertex but repeats a vertex whenever the tree branches; replacing such repeated stretches by direct edges can only shorten the tour because the triangle inequality makes the direct edge no longer than the two-leg path.

Exercise 3.1

3.1 *(Original.)* A roads inspector must travel along each road of the network with edge weights AB 6, AC 4, BC 5, CD 2, BD 7, AD 3 at least once, starting and ending at A.

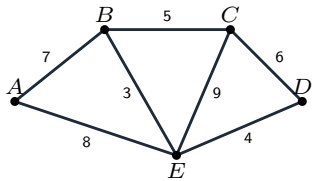
(a) Identify the vertices of odd degree. (b) List the possible pairings of the odd nodes with their total weights and choose the least. (c) Find the length of the inspector's shortest route. (d) State, with a reason, whether the route needs to repeat any roads. [9]



Exercise 3.1

3.1 *(Original.)* A roads inspector must travel along each road of the network with edge weights AB 6, AC 4, BC 5, CD 2, BD 7, AD 3 at least once, starting and ending at A.

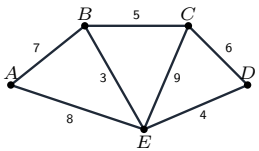
(a) Identify the vertices of odd degree. (b) List the possible pairings of the odd nodes with their total weights and choose the least. (c) Find the length of the inspector's shortest route. (d) State, with a reason, whether the route needs to repeat any roads. [9]



3.1 (a) Degrees: A 3 (AB, AC, AD), B 3 (AB, BC, BD), C 3 (AC, BC, CD), D 3 (CD, BD, AD) – all four vertices are odd. (b) Pairings: AB + CD = 6 + 2 = 8; AC + BD = 4 + 7 = 11; AD + BC = 3 + 5 = 8. Least = 8 (either AB + CD or AD + BC). (c) Total of all edges = 6 + 4 + 5 + 2 + 7 + 3 = 27; shortest route = 27 + 8 = 35. (d) Yes: four odd-degree vertices cannot form a closed trail using each edge once – exactly the chosen pairing's edges (AB and CD, or AD and BC) must be repeated.

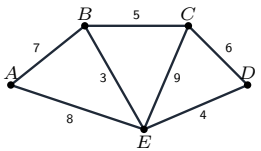
Exercise 3.2

3.2 *(Original.)* A sales representative must visit each of four cities J, K, L, M once and return to the start. The complete table of shortest distances is JK 8, JL 5, JM 9, KL 4, KM 10, LM 6. (a) Use the nearest neighbour algorithm from J to find a tour and state its length. (b) Find the minimum spanning tree of the network and use it to obtain a lower bound for the optimal tour by deleting J. (c) Write down an interval containing the optimal tour length. [9]



Exercise 3.2

3.2 *(Original.)* A sales representative must visit each of four cities J, K, L, M once and return to the start. The complete table of shortest distances is JK 8, JL 5, JM 9, KL 4, KM 10, LM 6. (a) Use the nearest neighbour algorithm from J to find a tour and state its length. (b) Find the minimum spanning tree of the network and use it to obtain a lower bound for the optimal tour by deleting J. (c) Write down an interval containing the optimal tour length. [9]



3.2 (a) From J: nearest L (5); from L: nearest K (4); from K: nearest M (10); return M–J (9). Tour J–L–K–M–J = 5 + 4 + 10 + 9 = 28. (b) MST (Kruskal): KL 4 ✓, JL 5 ✓, LM 6 ✓ (tree on J, K, L, M), total 15. Deleting J: MST of {K, L, M} = KL 4 + LM 6 = 10, plus J's two cheapest edges JL 5 + JK 8 = 13: lower bound = 10 + 13 = 23. (c) The optimal tour lies between 23 and 28 (the nearest-neighbour upper bound).

Local review guidance: odd nodes listed with degrees; ALL pairings costed before choosing; postman total = edges + chosen pairing; NN tour and the delete-a-vertex lower bound assembled from the MST of the rest plus the two cheapest at the deleted vertex. This is authored classroom guidance.

exercises.lol exercises.com.cn