Travelling salesman: upper bounds

Worked solutions

Further Maths topic practice

This topic online

1 A van driver must deliver to six shops in Cardiff, starting and finishing at the depot, and wants to keep the total distance as small as possible.

a. Explain the difference between the classical and the practical travelling salesman problems.

b. Explain why the length of any tour that visits every shop and returns to the depot is an upper bound for the length of the optimal route.

(3 marks)

a)

Classical: visit every vertex exactly once, then return to the start

This is a Hamiltonian cycle

Practical: visit every vertex at least once, then return to the start

A vertex may be passed through more than once

Classical exactly once; practical at least once

b)

The optimal route is the shortest of all the tours

So every tour has length $\geq$ the optimal length

Any tour length is an upper bound

Free for classroom and home use: print, photocopy and share. Not for resale or re-hosting.

2 The table shows the least distances, in miles, between five villages in Cumbria, A, B, C, D and E. A district nurse starts and finishes at A and visits every village.

	A	B	C	D	E
A	–	27	31	33	23
B	27	–	48	58	37
C	31	48	–	16	13
D	33	58	16	–	21
E	23	37	13	21	–

Use the nearest neighbour algorithm, starting at A, to find an upper bound for the length of her route. Write your tour and its length in the table below.

Nearest neighbour

Start	Tour	Length
A	A - E - C - D - B - A	137

(3 marks)

Nearest neighbour from A: each time go to the nearest vertex not yet visited

From A: E 23, B 27, C 31, D 33, nearest is E

From E: C 13, D 21, B 37, nearest is C

From C: D 16, B 48, nearest is D

From D: only B is left (58)

All vertices visited: return B to A (27)

Tour: A - E - C - D - B - A

Length = 23 + 13 + 16 + 58 + 27 = 137

A - E - C - D - B - A; upper bound 137 miles

Free for classroom and home use: print, photocopy and share. Not for resale or re-hosting.

3 The network below shows the direct roads between five farms, A to E. The number on each arc is the length of the road in km.

a. Complete the table of least distances below.

b. Use the nearest neighbour algorithm, starting at A, to find an upper bound for the length of a tour that visits every farm and returns to A.

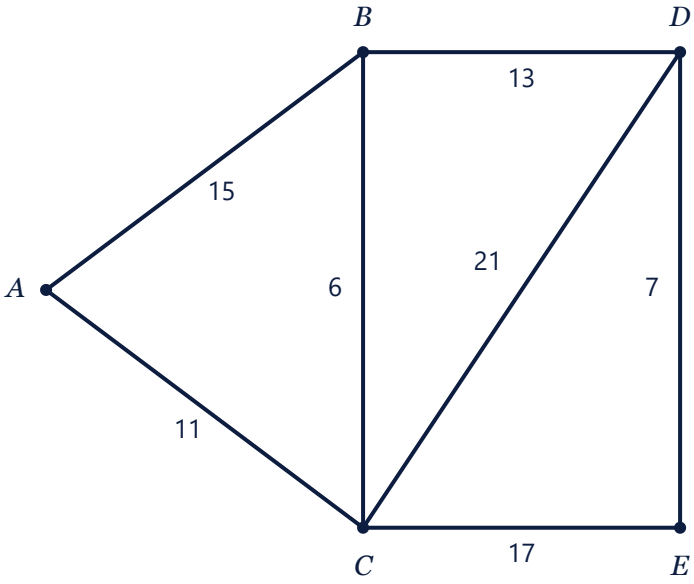


Table of least distances (km)

	A	B	C	D	E
A	-	15	11	28	28
B	15	-	6	13	20
C	11	6	-	19	17
D	28	13	19	-	7
E	28	20	17	7	-

Nearest neighbour

Start	Tour	Length
A	A - C - B - D - E - A	65

Diagram NOT accurately drawn

(4 marks)

a)

Least distance = shortest route, which may use other vertices

AD: A - B - D = 15 + 13 = 28

(A - C - D = 11 + 21 = 32 is longer)

AE: A - C - E = 11 + 17 = 28

BE: B - D - E = 13 + 7 = 20

Free for classroom and home use: print, photocopy and share. Not for resale or re-hosting.

CO: C - B - O = 6 + 13 = 19

This is less than the direct road CO = 21

AO 28, AE 28, BE 20, CO 19

b)

Nearest neighbour from A: each time go to the nearest vertex not yet visited

From A: C 11, B 15, O 28, E 28, nearest is C

From C: B 6, E 17, O 19, nearest is B

From B: O 13, E 20, nearest is O

From O: only E is left (7)

All vertices visited: return E to A (28)

Tour: A - C - B - O - E - A

Length = 11 + 6 + 13 + 7 + 28 = 65

A - C - B - O - E - A; upper bound 65 km

Free for classroom and home use: print, photocopy and share. Not for resale or re-hosting.

4 The table shows the least travel times, in minutes, between six depots, A to F, used by a parcel company in Kent.

	A	B	C	D	E	F
A	–	24	49	25	69	47
B	24	–	41	9	53	36
C	49	41	–	32	61	19
D	25	9	32	–	44	27
E	69	53	61	44	–	42
F	47	36	19	27	42	–

a. Use the nearest neighbour algorithm, starting at A, to find an upper bound for the length of a tour of the six depots.

b. Repeat the nearest neighbour algorithm, starting at D.

c. State the better upper bound, giving a reason for your answer.

Nearest neighbour tours

Start	Tour	Length
A	A - B - D - F - C - E - A	209
D	D - B - A - F - C - E - D	204

(5 marks)

a)

Nearest neighbour from A: each time go to the nearest vertex not yet visited

From A: B 24, D 25, F 47, C 49, E 69, nearest is B

From B: D 9, F 36, C 41, E 53, nearest is D

From D: F 27, C 32, E 44, nearest is F

From F: C 19, E 42, nearest is C

From C: only E is left (61)

All vertices visited: return E to A (69)

Tour: A - B - D - F - C - E - A

Length = 24 + 9 + 27 + 19 + 61 + 69 = 209

A - B - D - F - C - E - A, 209 minutes

b)

Nearest neighbour from D: each time go to the nearest vertex not yet visited

From D: B 9, A 25, F 27, C 32, E 44, nearest is B

From B: A 24, F 36, C 41, E 53, nearest is A

Free for classroom and home use: print, photocopy and share. Not for resale or re-hosting.

From A: F 47, C 49, E 69, nearest is F

From F: C 19, E 42, nearest is C

From C: only E is left (61)

All vertices visited: return E to O (44)

Tour: O - B - A - F - C - E - O

Length = 9 + 24 + 47 + 19 + 61 + 44 = 204

O - B - A - F - C - E - O, 204 minutes

c)

The optimal tour is at most 209 and at most 204

The smaller upper bound is closer to the optimal value

204 minutes

Free for classroom and home use: print, photocopy and share. Not for resale or re-hosting.

5 The network below shows the footpaths between six viewpoints, A to F, in a country park in Yorkshire. The numbers on the arcs are lengths in hundreds of metres. A ranger must visit every viewpoint, starting and finishing at A.

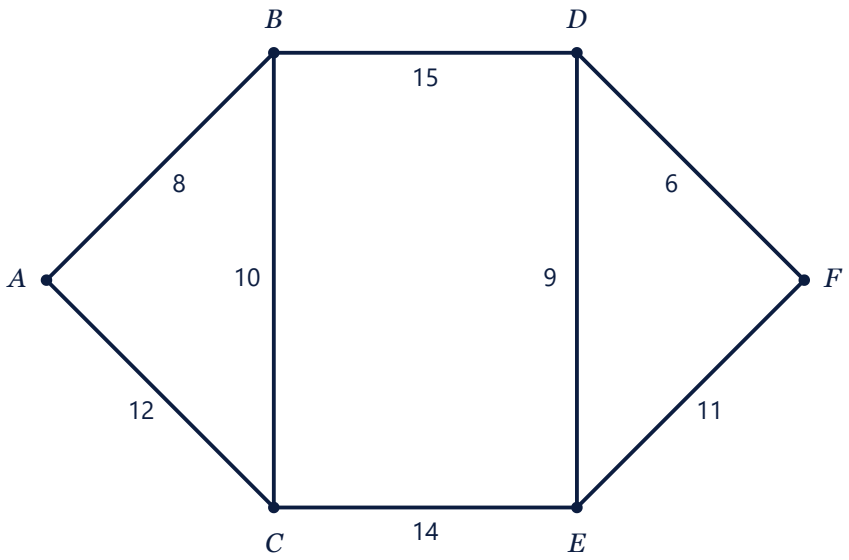
a. Apply the nearest neighbour algorithm, starting at A, directly to the network. Explain why it does not give a tour.

The table shows the least distances between the viewpoints.

	A	B	C	D	E	F
A	–	8	12	23	26	29
B	8	–	10	15	24	21
C	12	10	–	23	14	25
D	23	15	23	–	9	6
E	26	24	14	9	–	11
F	29	21	25	6	11	–

b. Use the nearest neighbour algorithm on the table, starting at A, to find an upper bound for the length of the ranger's route.

c. Write down the route the ranger actually walks in the network, listing every viewpoint passed through.



Answer

	Order of viewpoints	Length
Tour (b)	A - B - C - E - D - F - A	76
Route (c)	A - B - C - E - D - F - D - B - A	76

Diagram NOT accurately drawn

(7 marks)

a)

From A: B 8, C 12, nearest is B

From B: C 10, D 15, nearest is C

Free for classroom and home use: print, photocopy and share. Not for resale or re-hosting.

From C: only E is unvisited next to C (14)

From E: O 9, F 11, nearest is O

From O: F 6

All visited, but there is no arc F to A

The network is not complete, so we cannot return to A

No arc from F to A: the algorithm stalls at F

b)

Nearest neighbour from A: each time go to the nearest vertex not yet visited

From A: B 8, C 12, O 23, E 26, F 29, nearest is B

From B: C 10, O 15, F 21, E 24, nearest is C

From C: E 14, O 23, F 25, nearest is E

From E: O 9, F 11, nearest is O

From O: only F is left (6)

All vertices visited: return F to A (29)

Tour: A - B - C - E - O - F - A

Length = 8 + 10 + 14 + 9 + 6 + 29 = 76

A - B - C - E - O - F - A, upper bound 76 (7600 m)

c)

Each step uses a least distance: replace it with its path

A - B, B - C, C - E, E - O, O - F are arcs

F to A: F - O - B - A = 6 + 15 + 8 = 29

A - B - C - E - O - F - O - B - A

Free for classroom and home use: print, photocopy and share. Not for resale or re-hosting.

6 The table below shows the least distances, in km, between six sports centres, A to F, in a borough of London.

a. Use Prim's algorithm, starting at A, to find a minimum spanning tree for the table. You must state the order in which you select the arcs.

b. Hence find an initial upper bound for the length of a tour of the six centres.

c. Starting from the route that travels along every arc of the tree twice, use shortcuts to reduce the upper bound to below 160 km. State your tour and its length.

	1	2	3	5	4	6
	A	**B**	**C**	**D**	**E**	**F**
A	-	22	31	40	28	54
B	(22)	-	9	49	11	51
C	31	(9)	-	58	14	46
D	(40)	49	58	-	60	21
E	28	(11)	14	60	-	52
F	54	51	46	(21)	52	-

Upper bounds

Method	Tour	Length
2 × MST	A - D - F - D - A - B - C - B - E - B - A	206
Shortcuts	A - D - F - C - B - E - A	155

(7 marks)

a)

Start at A: delete row A, number column A as 1

Each time find the smallest undeleted entry in the numbered columns

Smallest is 22 in column A, row B: add AB

Delete row B, number column B as 2

Smallest is 9 in column B, row C: add BC

Delete row C, number column C as 3

Smallest is 11 in column B, row E: add BE

Delete row E, number column E as 4

Smallest is 40 in column A, row D: add AD

Delete row D, number column D as 5

Smallest is 21 in column D, row F: add DF

Delete row F, number column F as 6

Weight = 22 + 9 + 11 + 40 + 21 = 103

Free for classroom and home use: print, photocopy and share. Not for resale or re-hosting.

AB, BC, BE, AD, DF; weight 103 km

b)

Use every arc of the tree twice

Upper bound = 2 × 103 = 206

206 km

c)

Closed route on the tree from A:

A - D - F - D - A - B - C - B - E - B - A

Length = 2 × 103 = 206

Shortcut 1: go straight from F to C

F - D - A - B - C = 21 + 40 + 22 + 9 = 92

Saving = 92 - 46 = 46

Route A - D - F - C - B - E - B - A, length 160

Shortcut 2: go straight from E to A

E - B - A = 11 + 22 = 33

Saving = 33 - 28 = 5

Tour A - D - F - C - B - E - A

Length = 40 + 21 + 46 + 9 + 11 + 28 = 155

A - D - F - C - B - E - A, 155 km

Free for classroom and home use: print, photocopy and share. Not for resale or re-hosting.

7 The table shows the least distances, in miles, between six towns, A to F, in Lincolnshire. The least distance between D and F is x miles, where $x \geq 20$

	A	B	C	D	E	F
A	–	63	49	31	38	46
B	63	–	56	53	25	72
C	49	56	–	48	40	67
D	31	53	48	–	28	x
E	38	25	40	28	–	47
F	46	72	67	x	47	–

The nearest neighbour algorithm, starting at A, is applied to the table. The tour begins A, D, F.

a. Find the range of possible values of x

b. Complete the tour and find, in terms of x, the upper bound it gives.

c. Given instead that $x = 35$, use the nearest neighbour algorithm, starting at A, to find an upper bound.

Nearest neighbour from A

Part	Tour	Length
(b)	A - D - F - E - B - C - A	208 + x
(c)	A - D - E - B - C - F - A	253

(6 marks)

a)

From A: B 63, C 49, D 31, E 38, F 46, nearest is D

From D: B 53, C 48, E 28, F x

F is chosen, so x is the smallest

$x < 28$

Given $x \geq 20$

$20 \leq x < 28$

b)

From F: B 72, C 67, E 47, nearest is E

From E: B 25, C 40, nearest is B

From B: only C is left (56)

All vertices visited: return C to A (49)

Tour: A - D - F - E - B - C - A

Length = $31 + x + 47 + 25 + 56 + 49$

Length = 208 + x

A - D - F - E - B - C - A; upper bound 208 + x

c)

Nearest neighbour from A: each time go to the nearest vertex not yet visited

From A: D 31, E 38, F 46, C 49, B 63, nearest is D

From D: E 28, F 35, C 48, B 53, nearest is E

From E: B 25, C 40, F 47, nearest is B

From B: C 56, F 72, nearest is C

From C: only F is left (67)

All vertices visited: return F to A (46)

Tour: A - D - E - B - C - F - A

Length = 31 + 28 + 25 + 56 + 67 + 46 = 253

A - D - E - B - C - F - A; upper bound 253 miles

Free for classroom and home use: print, photocopy and share. Not for resale or re-hosting.

8 The table shows the least distances, in km, between six branches of a bank in Northern Ireland, A to F. An auditor visits every branch and returns to her starting point.

	A	B	C	D	E	F
A	–	37	60	52	51	46
B	37	–	59	49	39	30
C	60	59	–	10	22	29
D	52	49	10	–	24	23
E	51	39	22	24	–	19
F	46	30	29	23	19	–

a. Use the nearest neighbour algorithm, starting at C, to find an upper bound for the length of her tour.

Kofi suggests the tour A - B - F - E - C - D - A.

b. Find the length of Kofi's tour.

c. State the best upper bound you now have, and explain whether you can say that it is the length of the optimal tour.

Tours

Start	Tour	Length
C (NN)	C - D - F - E - B - A - C	188
Kofi	A - B - F - E - C - D - A	170

(6 marks)

a)

Nearest neighbour from C: each time go to the nearest vertex not yet visited

From C: D 10, E 22, F 29, B 59, A 60, nearest is D

From D: F 23, E 24, B 49, A 52, nearest is F

From F: E 19, B 30, A 46, nearest is E

From E: B 39, A 51, nearest is B

From B: only A is left (37)

All vertices visited: return A to C (60)

Tour: C - D - F - E - B - A - C

Length = 10 + 23 + 19 + 39 + 37 + 60 = 188

C - D - F - E - B - A - C, 188 km

b)

AB + BF + FE + EC + CD + DA = 37 + 30 + 19 + 22 + 10 + 52

Free for classroom and home use: print, photocopy and share. Not for resale or re-hosting.

Length = 170

170 km

c)

170 < 188, so 170 is the better upper bound

optimal tour ≤ 170

A tour only gives an upper bound

We would need a lower bound equal to 170 to know it is optimal

170 km; not known to be optimal

Free for classroom and home use: print, photocopy and share. Not for resale or re-hosting.

9 The network below shows the roads between seven villages, A to G, on Anglesey. The numbers on the arcs are distances in km. A mobile library must visit every village, starting and finishing at A.

a. Complete the table of least distances below.

b. Use the nearest neighbour algorithm on the completed table, starting at A, to find an upper bound for the length of the route.

c. Write down the route in the network that the library van actually drives for your tour in (b).

The nearest neighbour algorithm starting at G gives the tour G - E - D - B - A - C - F - G.

d. Find the length of this tour and state, with a reason, which upper bound is better.

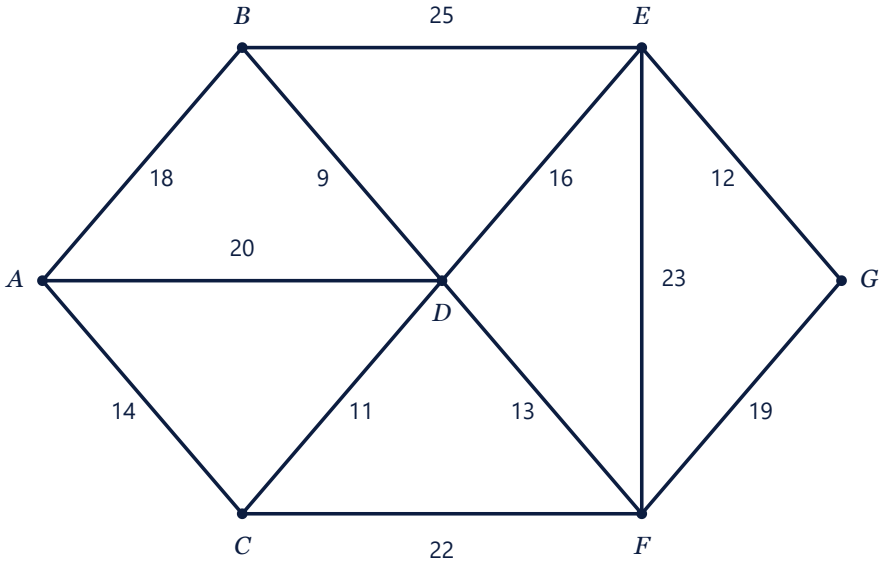


Table of least distances (km)

	A	B	C	D	E	F	G
A	-	18	14	20	36	33	48
B	18	-	20	9	25	22	37
C	14	20	-	11	27	22	39
D	20	9	11	-	16	13	28
E	36	25	27	16	-	23	12
F	33	22	22	13	23	-	19
G	48	37	39	28	12	19	-

Answer

	Order of villages	Length
Tour (b)	A - C - D - B - F - G - E - A	123
Route (c)	A - C - D - B - D - F - G - E - D - A	123
From G (d)	G - E - D - B - A - C - F - G	110

Diagram NOT accurately drawn

(9 marks)

a)

Find the shortest path for each blank by inspection

Free for classroom and home use: print, photocopy and share. Not for resale or re-hosting.

AG: A - D - E - G = 20 + 16 + 12 = 48

BG: B - E - G = 25 + 12 = 37

CE: C - D - E = 11 + 16 = 27

DG: D - E - G = 16 + 12 = 28

AG 48, BG 37, CE 27, DG 28

b)

Nearest neighbour from A: each time go to the nearest vertex not yet visited

From A: C 14, B 18, D 20, F 33, E 36, G 48, nearest is C

From C: D 11, B 20, F 22, E 27, G 39, nearest is D

From D: B 9, F 13, E 16, G 28, nearest is B

From B: F 22, E 25, G 37, nearest is F

From F: G 19, E 23, nearest is G

From G: only E is left (12)

All vertices visited: return E to A (36)

Tour: A - C - D - B - F - G - E - A

Length = 14 + 11 + 9 + 22 + 19 + 12 + 36 = 123

A - C - D - B - F - G - E - A, 123 km

c)

AC, CD, DB, FG, GE are arcs of the network

BF = 22: path B - D - F = 9 + 13

EA = 36: path E - D - A = 16 + 20

A - C - D - B - D - F - G - E - D - A

d)

Length = 12 + 16 + 9 + 18 + 14 + 22 + 19 = 110

110 < 123

110 km is the better upper bound

Free for classroom and home use: print, photocopy and share. Not for resale or re-hosting.

10 The table shows the least travel times, in minutes, between five care homes, A to E, in Leicester. A visiting doctor plans a tour of all five homes, returning to the first.

	A	B	C	D	E
A	–	29	11	22	33
B	29	–	27	35	57
C	11	27	–	14	30
D	22	35	14	–	43
E	33	57	30	43	–

a. Use the nearest neighbour algorithm, starting at each of the five homes in turn, to find five tours. State the length of each tour.

b. Hence state the best upper bound for the length of the optimal tour.

c. Explain why the doctor may start her tour at any of the homes.

Nearest neighbour tours

Start	Tour	Length
A	A - C - D - B - E - A	150
B	B - C - A - D - E - B	160
C	C - A - D - B - E - C	155
D	D - C - A - B - E - D	154
E	E - C - A - D - B - E	155

(7 marks)

a)

Nearest neighbour: each time go to the nearest home not yet visited

Start at A

From A: C 11, D 22, B 29, E 33, nearest is C

From C: D 14, B 27, E 30, nearest is D

From D: B 35, E 43, nearest is B

From B: only E is left (57)

All vertices visited: return E to A (33)

Tour: A - C - D - B - E - A

Length = 11 + 14 + 35 + 57 + 33 = 150

Start at B

From B: C 27, A 29, D 35, E 57, nearest is C

From C: A 11, D 14, E 30, nearest is A

Free for classroom and home use: print, photocopy and share. Not for resale or re-hosting.

From A: D 22, E 33, nearest is D

From D: only E is left (43)

All vertices visited: return E to B (57)

Tour: B - C - A - D - E - B

Length = 27 + 11 + 22 + 43 + 57 = 160

Start at C

From C: A 11, D 14, B 27, E 30, nearest is A

From A: D 22, B 29, E 33, nearest is D

From D: B 35, E 43, nearest is B

From B: only E is left (57)

All vertices visited: return E to C (30)

Tour: C - A - D - B - E - C

Length = 11 + 22 + 35 + 57 + 30 = 155

Start at D

From D: C 14, A 22, B 35, E 43, nearest is C

From C: A 11, B 27, E 30, nearest is A

From A: B 29, E 33, nearest is B

From B: only E is left (57)

All vertices visited: return E to D (43)

Tour: D - C - A - B - E - D

Length = 14 + 11 + 29 + 57 + 43 = 154

Start at E

From E: C 30, A 33, D 43, B 57, nearest is C

From C: A 11, D 14, B 27, nearest is A

From A: D 22, B 29, nearest is D

From D: only B is left (35)

All vertices visited: return B to E (57)

Tour: E - C - A - D - B - E

Length = 30 + 11 + 22 + 35 + 57 = 155

Free for classroom and home use: print, photocopy and share. Not for resale or re-hosting.

150, 160, 155, 154, 155 minutes

b)

Best upper bound = smallest tour length

150 minutes

c)

A tour is a cycle through all five homes

The same cycle can start at any home and has the same length

Any starting home gives the same cycle length

Free for classroom and home use: print, photocopy and share. Not for resale or re-hosting.

11 The table shows the least distances, in km, between six warehouses, A to F, near Birmingham. The least distance between B and D is x km, where $25 < x < 63$

	A	B	C	D	E	F
A	–	54	50	60	85	59
B	54	–	20	x	35	26
C	50	20	–	44	40	9
D	60	x	44	–	42	47
E	85	35	40	42	–	49
F	59	26	9	47	49	–

a. Given that $x < 35$, use the nearest neighbour algorithm, starting at A, to find an upper bound, in terms of x, for the length of a tour of the warehouses.

b. Given instead that $x > 35$, use the nearest neighbour algorithm, starting at A, to find an upper bound.

c. Find the set of values of x for which the nearest neighbour algorithm, starting at A, gives an upper bound of less than 240 km.

Nearest neighbour from A

Case	Tour	Length
x < 35	A - C - F - B - D - E - A	212 + x
x > 35	A - C - F - B - E - D - A	222

(8 marks)

a)

From A: B 54, C 50, D 60, E 85, F 59, nearest is C

From C: B 20, D 44, E 40, F 9, nearest is F

From F: B 26, D 47, E 49, nearest is B

From B: D x, E 35

$x < 35$, so nearest is D

From D: only E is left (42)

All vertices visited: return E to A (85)

Tour: A - C - F - B - D - E - A

Length = $50 + 9 + 26 + x + 42 + 85 = 212 + x$

A - C - F - B - D - E - A; upper bound $212 + x$

b)

From B: D x, E 35

Free for classroom and home use: print, photocopy and share. Not for resale or re-hosting.

$x > 35$, so nearest is E

From E: only O is left (42)

All vertices visited: return O to A (60)

Tour: A - C - F - B - E - O - A

Length = $50 + 9 + 26 + 35 + 42 + 60 = 222$

222 km

c)

Case $25 < x < 35$

$212 + x < 240$

$x < 28$

Case $35 < x < 63$

$222 < 240$ for all these x

($x = 35$ gives a tie at B, so leave it out)

$25 < x < 28$ or $35 < x < 63$

Free for classroom and home use: print, photocopy and share. Not for resale or re-hosting.

12 A tour guide in Edinburgh plans a walking tour of seven sites, A to G, starting and finishing at the hotel at A. The table below shows the least walking times, in minutes, between the sites.

a. Use Prim's algorithm, starting at A, to find a minimum spanning tree. You must state the order in which you select the arcs, and its weight.

b. Hence find an initial upper bound for the length of the tour.

c. Use a shortcut to reduce this upper bound to below 140 minutes. State the tour you use.

d. Use the nearest neighbour algorithm, starting at A, to find a second upper bound.

e. State, with a reason, the best upper bound you have found.

	1	5	6	2	7	3	4
	A	**B**	**C**	**D**	**E**	**F**	**G**
A	-	37	52	12	43	21	29
B	37	-	18	34	24	32	10
C	52	18	-	40	14	50	28
D	12	34	40	-	31	27	30
E	43	24	14	31	-	42	33
F	21	32	50	27	42	-	22
G	29	10	28	30	33	22	-

Upper bounds

Method	Tour	Length
2 × MST		194
Shortcut	A - F - G - B - C - E - D - A	128
NN from A	A - D - F - G - B - C - E - A	146

(11 marks)

a)

Start at A: delete row A, number column A as 1

Each time find the smallest undeleted entry in the numbered columns

Smallest is 12 in column A, row D: add AD

Delete row D, number column D as 2

Smallest is 21 in column A, row F: add AF

Delete row F, number column F as 3

Smallest is 22 in column F, row G: add FG

Delete row G, number column G as 4

Smallest is 10 in column G, row B: add GB

Free for classroom and home use: print, photocopy and share. Not for resale or re-hosting.

Delete row B, number column B as 5

Smallest is 18 in column B, row C: add BC

Delete row C, number column C as 6

Smallest is 14 in column C, row E: add CE

Delete row E, number column E as 7

Weight = 12 + 21 + 22 + 10 + 18 + 14 = 97

AO, AF, FG, GB, BC, CE; weight 97

b)

Upper bound = 2 × 97 = 194

194 minutes

c)

The tree is the path O - A - F - G - B - C - E

Closed route: A - F - G - B - C - E - C - B - G - F - A - O - A

Length = 2 × 97 = 194

Shortcut: go straight from E to O

E - C - B - G - F - A - O = 14 + 18 + 10 + 22 + 21 + 12 = 97

Saving = 97 - 31 = 66

New upper bound = 194 - 66 = 128

Tour: A - F - G - B - C - E - O - A

Check: 21 + 22 + 10 + 18 + 14 + 31 + 12 = 128

A - F - G - B - C - E - O - A, 128 minutes

d)

Nearest neighbour from A: each time go to the nearest vertex not yet visited

From A: O 12, F 21, G 29, B 37, E 43, C 52, nearest is O

From O: F 27, G 30, E 31, B 34, C 40, nearest is F

From F: G 22, B 32, E 42, C 50, nearest is G

From G: B 10, C 28, E 33, nearest is B

From B: C 18, E 24, nearest is C

From C: only E is left (14)

Free for classroom and home use: print, photocopy and share. Not for resale or re-hosting.

All vertices visited: return E to A (43)

Tour: A - D - F - G - B - C - E - A

Length = 12 + 27 + 22 + 10 + 18 + 14 + 43 = 146

A - D - F - G - B - C - E - A, 146 minutes

e)

128 < 146 < 194

The smallest upper bound is the best

128 minutes

Free for classroom and home use: print, photocopy and share. Not for resale or re-hosting.