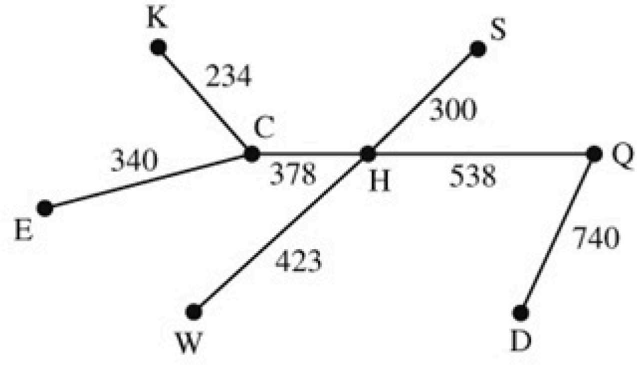
Chapter Review 3

1 a i Arcs are labelled with initial letters of the nodes.
CK add to tree
SH add to tree
CE add to tree
EK reject
CH add to tree
HW add to tree
CS reject
HQ add to tree
QS reject
QD add to tree
KS reject
DW reject
EW reject

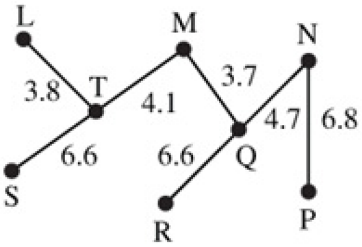
ii *EC*
CK
CH
HS
HW
HQ
QD

b



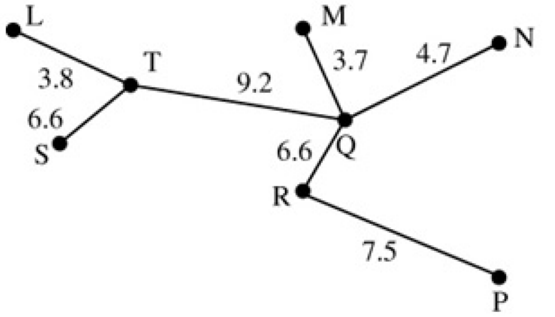
weight: 2953

2 a i *LT*
MT
MQ
NQ
ST
QR
NP



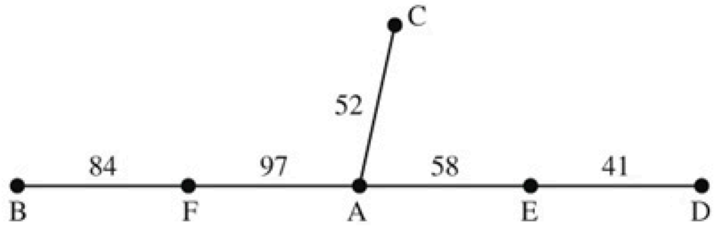
ii *MQ* (3.7) add to tree
LT (3.8) add to tree
MT (4.1) add to tree
NQ (4.7) add to tree
MN (5.3) add to tree
ST (6.6) add to tree
QR (6.6) add to tree
NP (6.8) add to tree
reject remaining arcs

2 **b** Start off the tree with *QT* and *PR* then apply Kruskal’s algorithm.
Prim’s algorithm requires the ‘growing’ tree to be connected at all times. When using Kruskal’s algorithm the tree can be built from non-connected sub-trees.



3 **a** Arcs in order:
AC
AE
DE
AF
BF

b

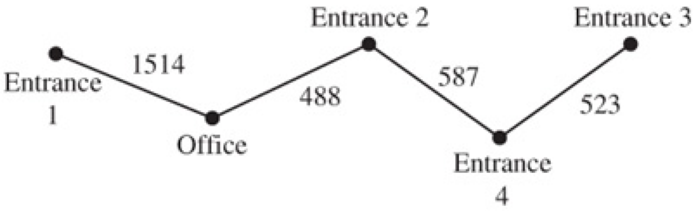


Length 332 mm

4 **a** Arcs in order:
Entrance 2 – Office
Entrance 2 – Entrance 4
Entrance 4 – Entrance 3
Office – Entrance 1

	↓2	↓5	↓1	↓4	↓3
	Office	Entrance 1	Entrance 2	Entrance 3	Entrance 4
Office	-	1514	488	980	945
Entrance 1	1514	-	1724	2446	2125
Entrance 2	488	1724	-	884	587
Entrance 3	980	2446	884	-	523
Entrance 4	945	2125	587	523	-

b



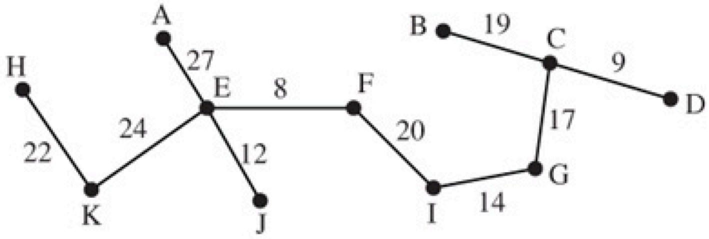
Length: 3112 m

5 a

29	27	30	19	9	26	17	18	8	12	24	20	14	14	22	26
8	29	27	30	19	9	26	17	18	12	24	20	14	14	22	26
8	9	17	12	14	14	18	29	27	30	19	26	24	20	22	26
8	9	12	17	14	14	18	19	24	20	22	26	29	27	30	26
8	9	12	14	17	14	18	19	20	24	22	26	29	27	26	30
8	9	12	14	14	17	18	19	20	22	24	26	26	27	29	30
EF	CD	EJ	FJ	GI	CG	DG	BC	FI	HK	EK	CF	JK	AE	AB	AH

b Order arcs into ascending order of weight and select the arc of least weight to start the tree: *EF* Consider the next arc of least weight, if it would form a cycle with the arcs already selected, reject it. Continue to select an arc of least weight until all vertices are connected to give a minimum spanning tree.

c



weight: 172

d $e = v - 1$

6 a Order of arcs

AC (1) add to tree

AD (2) add to tree

CD (2) reject

CF (3) add to tree

FG (3) add to tree

HI (3) add to tree

KM (3) add to tree

JK (3) add to tree

AB (4) add to tree

DE (4) add to tree

IJ (4) add to tree

LN (4) add to tree

DG (5) reject

BE (5) reject

IL (5) add to tree

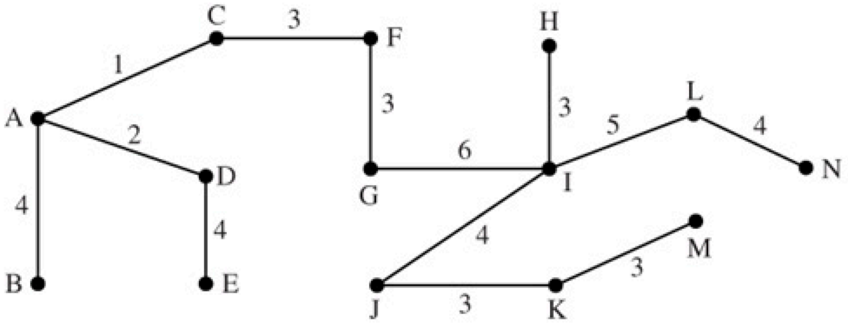
MN (5) reject

EG (6) reject

GI (6) add to tree

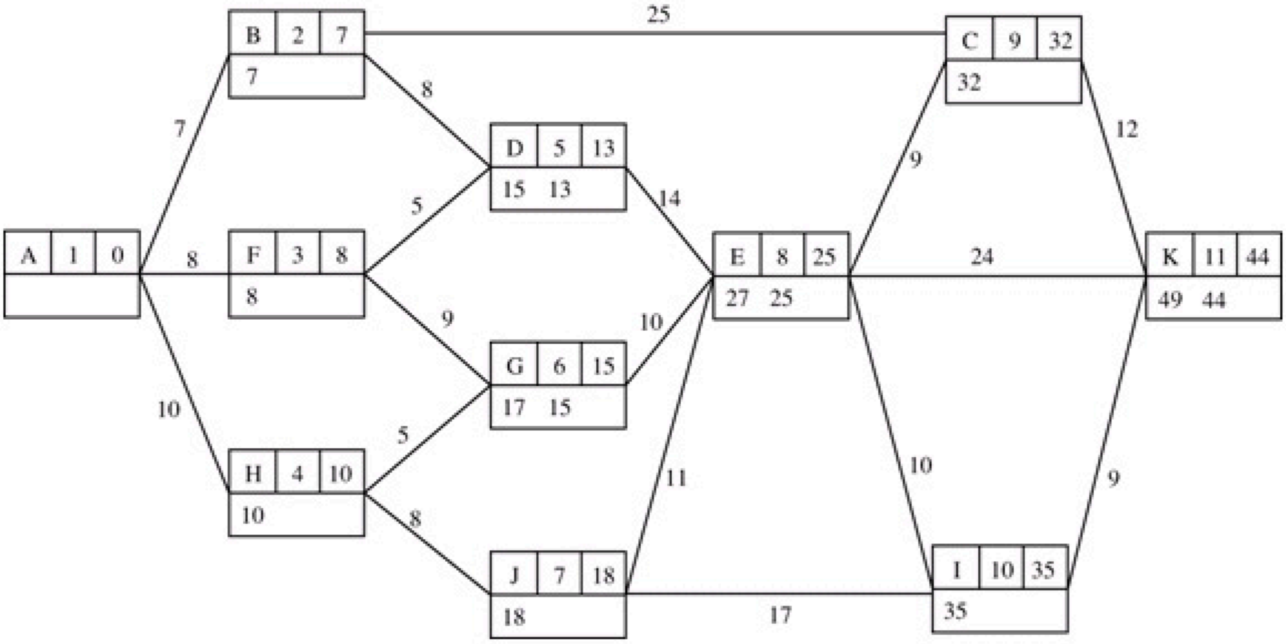
IM (6)
FH (7)
HL (7)
EJ (7)
reject remaining arcs

Weight = 45 so 4500 m needed



b Remove *FG* (7) and replace with *DG* (5) weight = 47 so 4700 m

7



a **i** Possible paths are *AHGEIK* and *AHJIK* and *ABCK* } Any one accepted

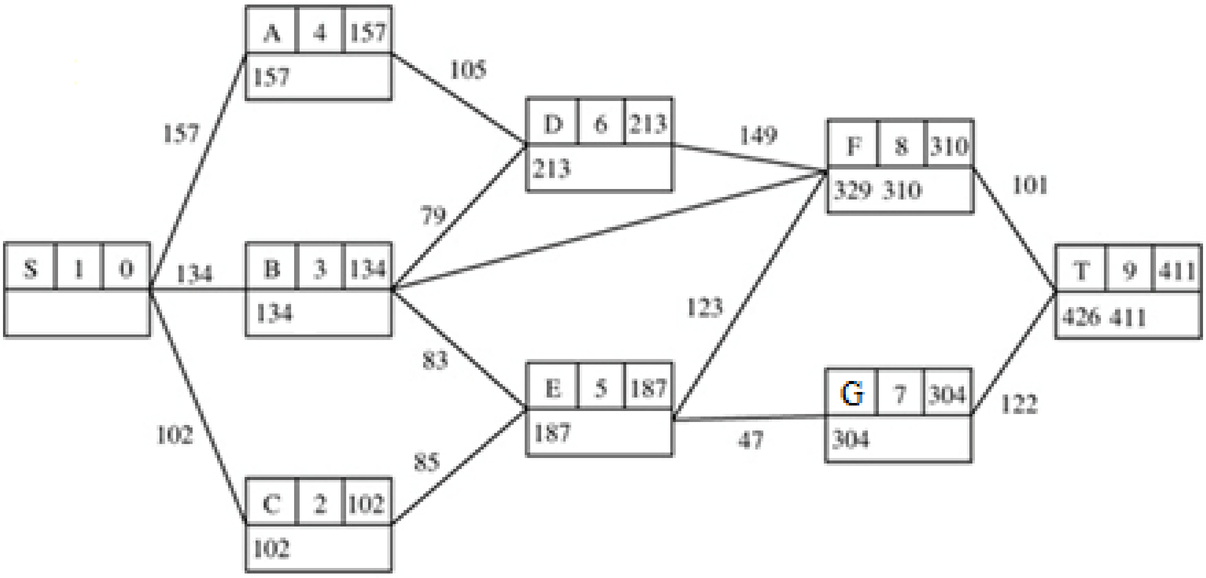
ii

44 – 9 = 35	*IK*		44 – 9 = 35	*IK*		44 – 12 = 32	*CK*
35 – 10 = 25	*EI*		35 – 17 = 18	*JI*		32 – 25 = 7	*BC*
25 – 10 = 15	*GE*	or	18 – 8 = 10	*MJ*	or	7 – 7 = 0	*AB*
15 – 5 = 10	*HG*		10 – 10 = 0	*AH*			
10 – 10 = 0	*AH*						

b *AHGEIK* and *AHJIK* and *ABCK*

c The arcs could be roads.
The nodes could be junctions.
The number on each arc could be the distance in km.
The network, together with Dijkstra’s algorithm, could be used to find the shortest route from *A* to *K*.

8

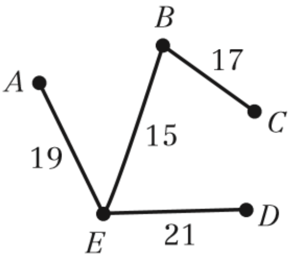


Order of vertex labelling: *SCBAEDGFT*

Route: *SCEFT*

411 – 101 = 310	*FT*
310 – 123 = 187	*EF*
187 – 85 = 102	*CE*
102 – 102 = 0	*SC*

9 a i, ii



Total weight = 72

b Prim’s algorithm grows a minimum spanning tree by adding one vertex at a time. The next vertex chosen to be added is always the shortest edge from the vertex already on the graph.
Kruskal's algorithm grows a minimum spanning tree by adding one edge at a time. The edge with the least weight is always the next to be added only if it does not create a cycle.

c Prim’s algorithm may be quicker on a graph with a large number of arcs, such as a complete network, as Kruskal’s algorithm would require arcs to be sorted by weight.

10 a Using Prim's algorithm
Starting at *A*

	1	5	3	2	6	8	4	7
	A	*B*	*C*	*D*	*E*	*F*	*G*	*H*
A	–	~~1970~~	~~1450~~	~~1290~~	~~2130~~	~~2010~~	~~1770~~	~~2470~~
B	~~1970~~	–	~~1390~~	~~2850~~	~~1110~~	~~3620~~	440	~~1790~~
C	1450	~~1390~~	–	~~1680~~	~~2280~~	~~3460~~	~~960~~	~~2890~~
D	1290	~~2850~~	~~1680~~	–	~~3330~~	~~2670~~	~~2500~~	~~3740~~
E	~~2130~~	1110	~~2280~~	~~3330~~	–	~~3160~~	~~1480~~	~~680~~
F	2010	~~3620~~	~~3460~~	~~2670~~	~~3160~~	–	~~3590~~	~~3020~~
G	~~1770~~	~~440~~	960	~~2500~~	~~1480~~	~~3590~~	–	~~2150~~
H	~~2470~~	~~1790~~	~~2890~~	~~3740~~	680	~~3020~~	~~2150~~	–

The first arc is *AD* (1290 km).
The second arc is *AC* (1450 km).
The third arc is *CG* (960 km).
The fourth arc is *GB* (440 km).
The fifth arc is *BE* (1110 km).
The sixth arc is *EH* (680 km).
The seventh arc is *AF* (2010 km).
The weight of the spanning tree is 7940 km.

10 b Using the nearest neighbour algorithm
Starting at *C*

	6	3	1	7	4	8	2	5
	A	*B*	*C*	*D*	*E*	*F*	*G*	*H*
A	–	~~1970~~	~~1450~~	~~1290~~	~~2130~~	~~2010~~	~~1770~~	2470
B	~~1970~~	–	~~1390~~	~~2850~~	~~1110~~	~~3620~~	440	~~1790~~
C	~~1450~~	~~1390~~	–	~~1680~~	~~2280~~	~~3460~~	~~960~~	~~2890~~
D	1290	~~2850~~	~~1680~~	–	~~3330~~	~~2670~~	~~2500~~	~~3740~~
E	~~2130~~	1110	~~2280~~	~~3330~~	–	~~3160~~	~~1480~~	~~680~~
F	~~2010~~	~~3620~~	~~3460~~	2670	~~3160~~	–	~~3590~~	~~3020~~
G	~~1770~~	~~440~~	960	~~2500~~	~~1480~~	~~3590~~	–	~~2150~~
H	~~2470~~	~~1790~~	~~2890~~	~~3740~~	680	~~3020~~	~~2150~~	–

The first arc is *CG* (960 km).
The second arc is *GB* (440 km).
The third arc is *BE* (1110 km).
The fourth arc is *EH* (680 km).
The fifth arc is *HA* (2470 km).
The sixth arc is *AD* (1290 km).
The seventh arc is *DF* (2670 km).
The shortest route is *CGBEHADF* which has weight 9620 km

c MST × 2 = 7940 × 2 = 15 880 km
Nearest neighbour and return = 9620 + 3460 = 13 080 km
The method using the nearest neighbour algorithm will give the shortest way.