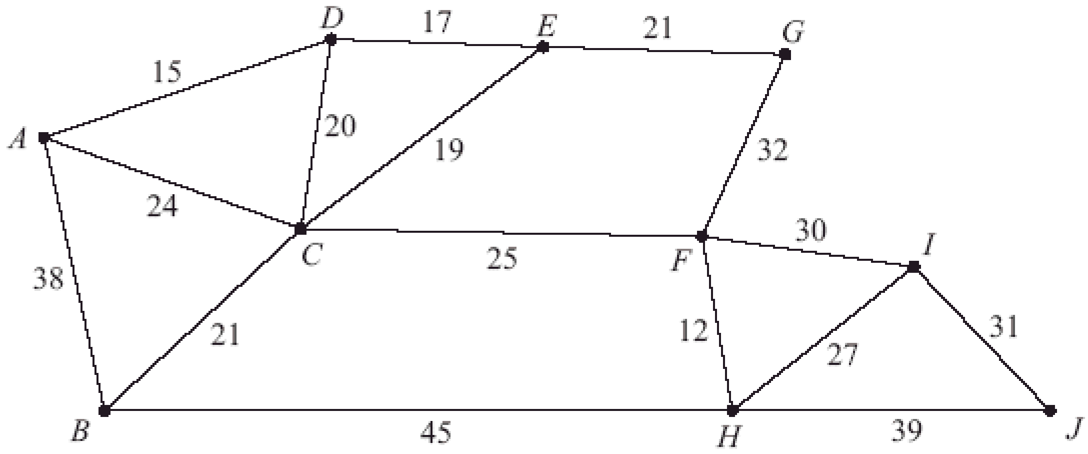
1.



The network in the diagram above shows the distances, in metres, between 10 wildlife observation points. The observation points are to be linked by footpaths, to form a network along the arcs indicated, using the least possible total length.

(a) Find a minimum spanning tree for the network in the diagram above, showing clearly the order in which you selected the arcs for your tree, using

(i) Kruskal's algorithm,

(3)

(ii) Prim's algorithm, starting from *A*.

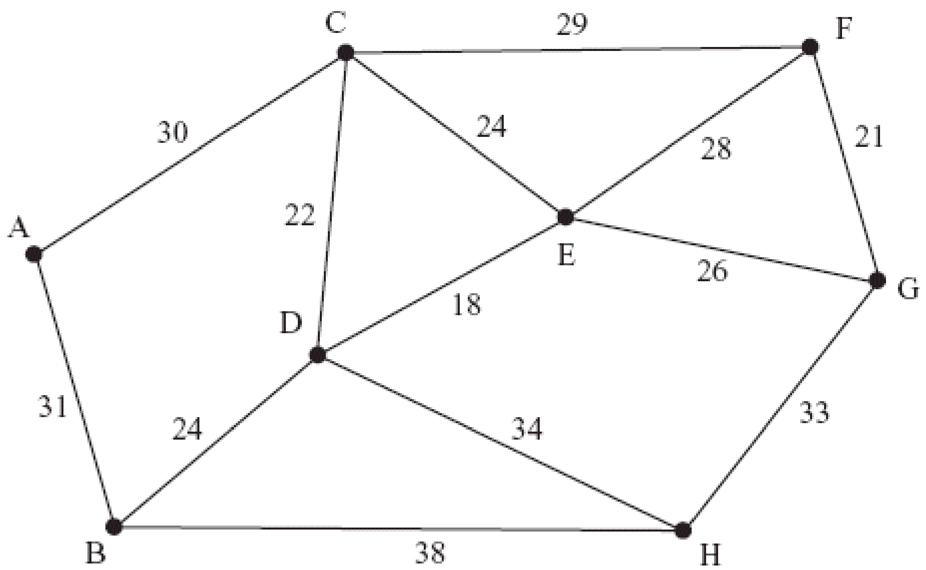
(3)

Given that footpaths are already in place along *AB* and *FI* and so should be included in the spanning tree,

(b) explain which algorithm you would choose to complete the tree, and how it should be adapted.
(You do **not** need to find the tree.)

(2)
(Total 8 marks)

2.



The diagram above represents the distances, in metres, between eight vertices, A, B, C, D, E, F, G and H, in a network.

(a) Use Kruskal's algorithm to find a minimum spanning tree for the network.
You should list the arcs in the order in which you consider them. In each case, state whether you are adding the arc to your minimum spanning tree.

(3)

(b) Complete the matrix below, to represent the network.

	A	B	C	D	E	F	G	H
A	–	31	30	–	–	–	–	–
B	31	–	–		–	–	–	38
C	30	–	–		24		–	–
D	–			–	18	–	–	
E	–	–	24	18	–	28		–
F	–	–		–	28	–	21	–
G	–	–	–	–		21	–	
H	–	38	–		–	–		–

(2)

(c) **Starting at A, use Prim's algorithm** to determine a minimum spanning tree. You must clearly state the order in which you considered the vertices and the order in which you included the arcs.

(3)

(d) State the weight of the minimum spanning tree.

(1)
(Total 9 marks)

3. Prim's algorithm finds a minimum spanning tree for a connected graph.

(a) Explain the terms

(i) connected graph,

(ii) tree,

(iii) spanning tree.

(3)

(b) Name an alternative algorithm for finding a minimum spanning tree.

(1)

	Cambridge	London	Norwich	Oxford	Portsmouth	Salisbury	York
Cambridge (C)	–	60	62	81	132	139	156
London (L)	60	–	116	56	74	88	211
Norwich (N)	62	116	–	144	204	201	181
Oxford (O)	81	56	144	–	84	63	184
Portsmouth (P)	132	74	204	84	–	43	269
Salisbury (S)	139	88	201	63	43	–	248
York (Y)	156	211	181	184	269	248	–

Figure 1

Figure 1 shows the distances by road, in miles, between seven cities.

(c) (i) Use Prim's algorithm, starting at London, to find the minimum spanning tree for these cities. You must clearly state the order in which you selected the edges of your tree, and the weight of the final tree.

(ii) Draw your tree using the vertices given in Figure 2 below.

Figure 2

(5)
(Total 9 marks)

4.

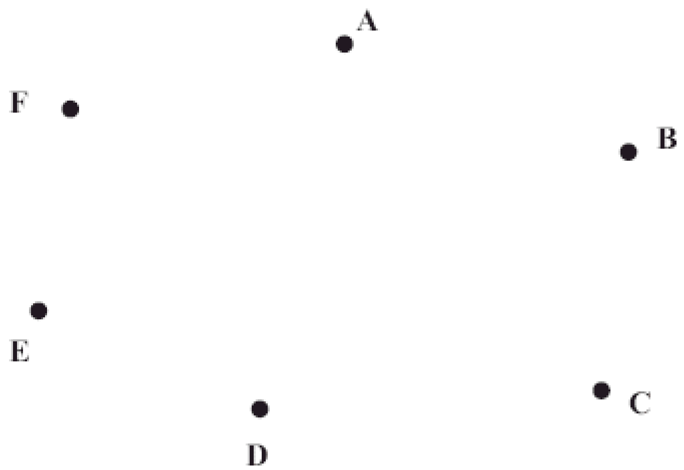
	A	**B**	**C**	**D**	**E**	**F**
A	–	135	180	70	95	225
B	135	–	215	125	205	240
C	180	215	–	150	165	155
D	70	125	150	–	100	195
E	95	205	165	100	–	215
F	225	240	155	195	215	–

The table shows the lengths, in km, of potential rail routes between six towns, A, B, C, D, E and F.

(a) Use Prim's algorithm, starting from A, to find a minimum spanning tree for this table. You must list the **arcs** that form your tree **in the order that they are selected**.

(3)

(b) Draw your tree using the vertices given in the diagram below.

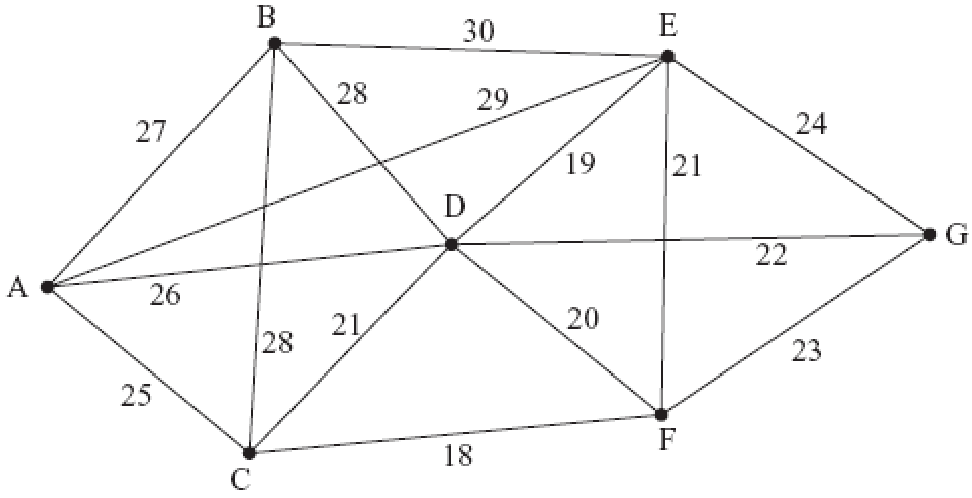


(1)

(c) State the total weight of your tree.

(1)
(Total 5 marks)

5.



(a) State two differences between Kruskal's algorithm and Prim's algorithm for finding a minimum spanning tree.

(2)

(b) Listing the arcs in the order that you consider them, find a minimum spanning tree for the network in the diagram above, using

(i) Prim's algorithm,

(ii) Kruskal's algorithm.

(6)
(Total 8 marks)

6.

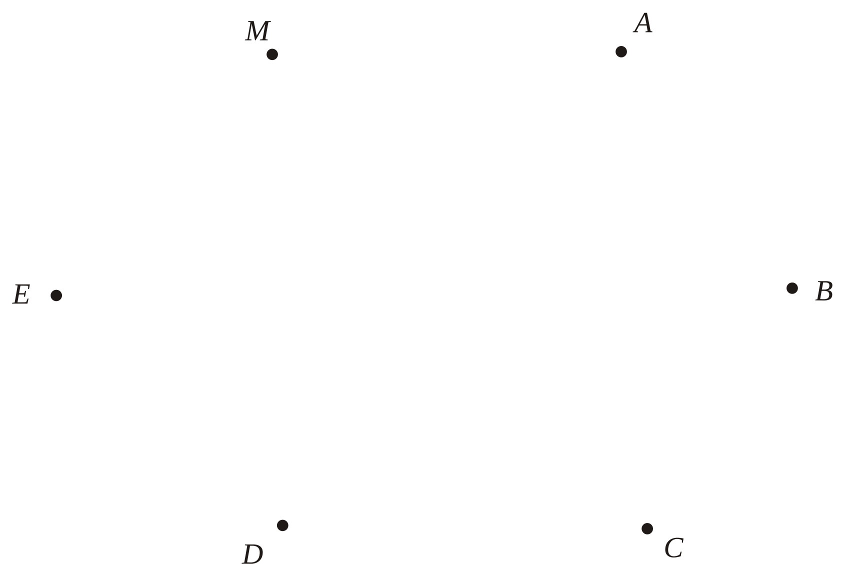
	M	*A*	*B*	*C*	*D*	*E*
M	–	215	170	290	210	305
A	215	–	275	100	217	214
B	170	275	–	267	230	200
C	290	100	267	–	180	220
D	210	217	230	180	–	245
E	305	214	200	220	245	–

The table shows the cost, in pounds, of linking five automatic alarm sensors, *A*, *B*, *C*, *D* and *E*, and the main reception, *M*.

(a) Use Prim's algorithm, starting from *M*, to find a minimum spanning tree for this table of costs. You must list the **arcs** that form your tree **in the order that they are selected**.

(3)

(b) Draw your tree using the vertices given in the diagram below.



(1)

(c) Find the total weight of your tree.

(1)

(d) Explain why it is not necessary to check for cycles when using Prim's algorithm.

(2)

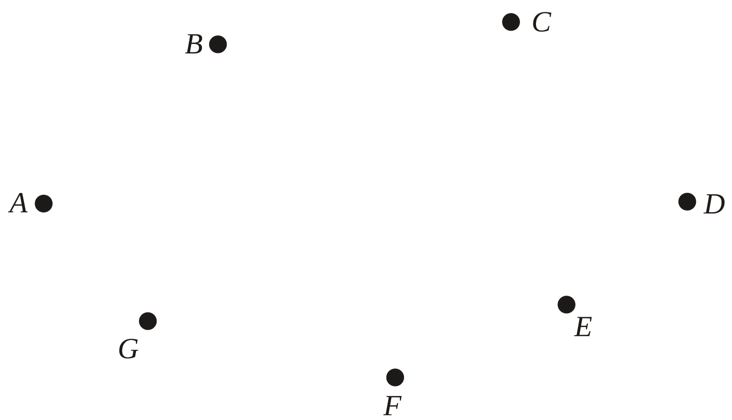
(Total 7 marks)

7.

	A	*B*	*C*	*D*	*E*	*F*	*G*
A	—	48	117	92	—	—	—
B	48	—	—	—	—	63	55
C	117	—	—	28	—	—	85
D	92	—	28	—	58	132	—
E	—	—	—	58	—	124	—
F	—	63	—	132	124	—	—
G	—	55	85	—	—	—	—

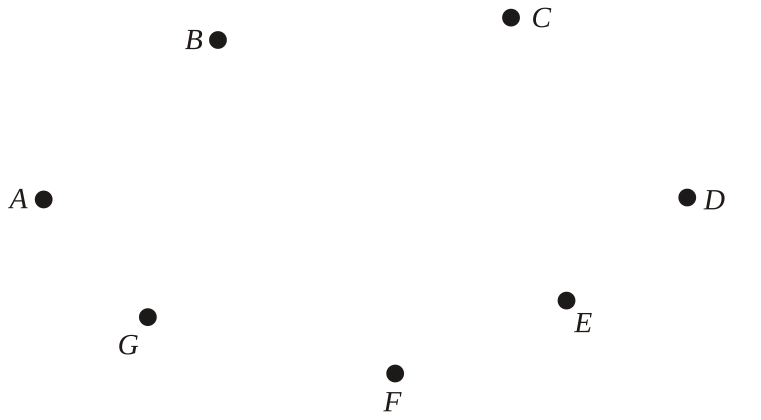
The table shows the lengths, in metres, of the paths between seven vertices *A*, *B*, *C*, *D*, *E*, *F* and *G* in a network N.

(a) Use Prim's algorithm, starting at *A*, to solve the minimum connector problem for this table of distances. You must clearly state the order in which you selected the edges of your tree, and the weight of your final tree. Draw your tree using the vertices given in the diagram below.



(5)

(b) Draw N using the vertices given in the diagram below.



(3)

(c) Solve the Route Inspection problem for N. You must make your method and working clear. State a shortest route and find its length.

(The weight of N is 802)

Shortest route:

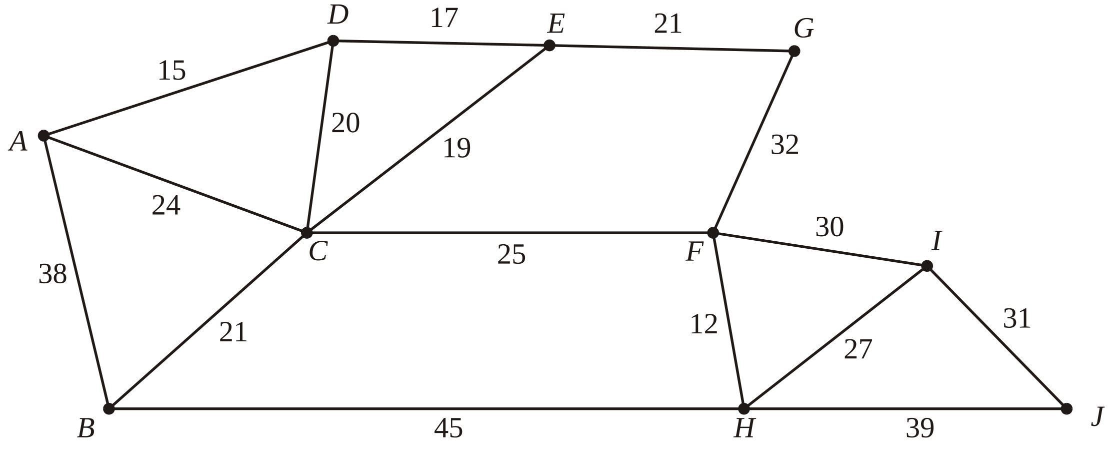
....

Length:

....

(7)
(Total 15 marks)

8.



The network in the diagram above shows the distances, in metres, between 10 wildlife observation points. The observation points are to be linked by footpaths, to form a network along the arcs indicated, using the least possible total length.

(a) Find a minimum spanning tree for the network in the diagram above, showing clearly the order in which you selected the arcs for your tree, using

(i) Kruskal's algorithm,

(3)

(ii) Prim's algorithm, starting from *A*.

(3)

Given that footpaths are already in place along *AB* and *FI* and so should be included in the spanning tree,

(b) explain which algorithm you would choose to complete the tree, and how it should be adapted. (You do **not** need to find the tree.)

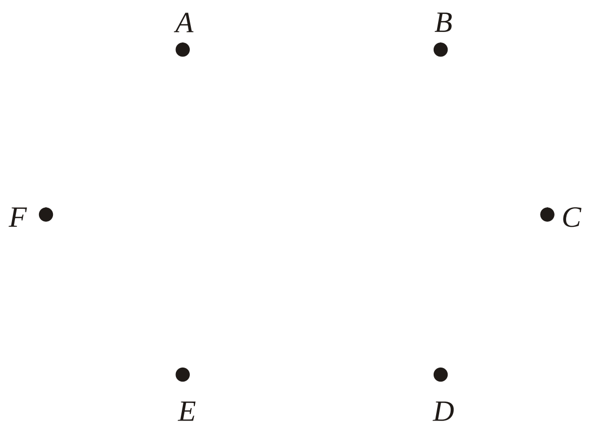
(2)
(Total 8 marks)

9.

	A	*B*	*C*	*D*	*E*	*F*
A	–	7	3	–	8	11
B	7	–	4	2	–	7
C	3	4	–	5	9	–
D	–	2	5	–	6	3
E	8	–	9	6	–	–
F	11	7	–	3	–	–

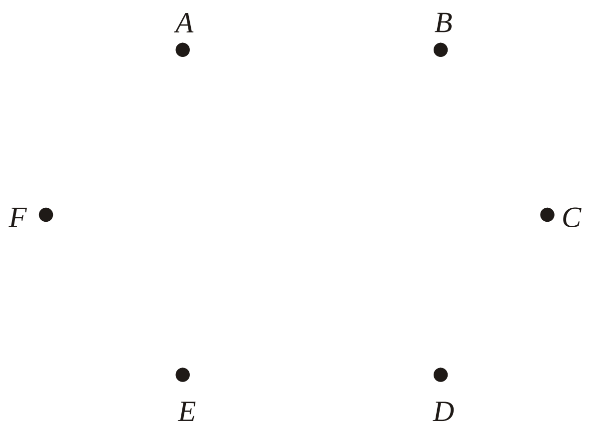
The matrix represents a network of roads between six villages *A*, *B*, *C*, *D*, *E* and *F*. The value in each cell represents the distance, in km, along these roads.

(a) Show this information on the diagram below.



(2)

(b) Use Kruskal's algorithm to determine the minimum spanning tree. State the order in which you include the arcs and the length of the minimum spanning tree. Draw the minimum spanning tree.



(5)

(c) Starting at *D*, use Prim's algorithm on the matrix given below to find the minimum spanning tree. State the order in which you include the arcs.

	A	*B*	*C*	*D*	*E*	*F*
A	–	7	3	–	8	11
B	7	–	4	2	–	7
C	3	4	–	5	9	–
D	–	2	5	–	6	3
E	8	–	9	6	–	–
F	11	7	–	3	–	–

(3)
(Total 10 marks)

10. (a) Define the terms

(i) tree,

(ii) spanning tree,

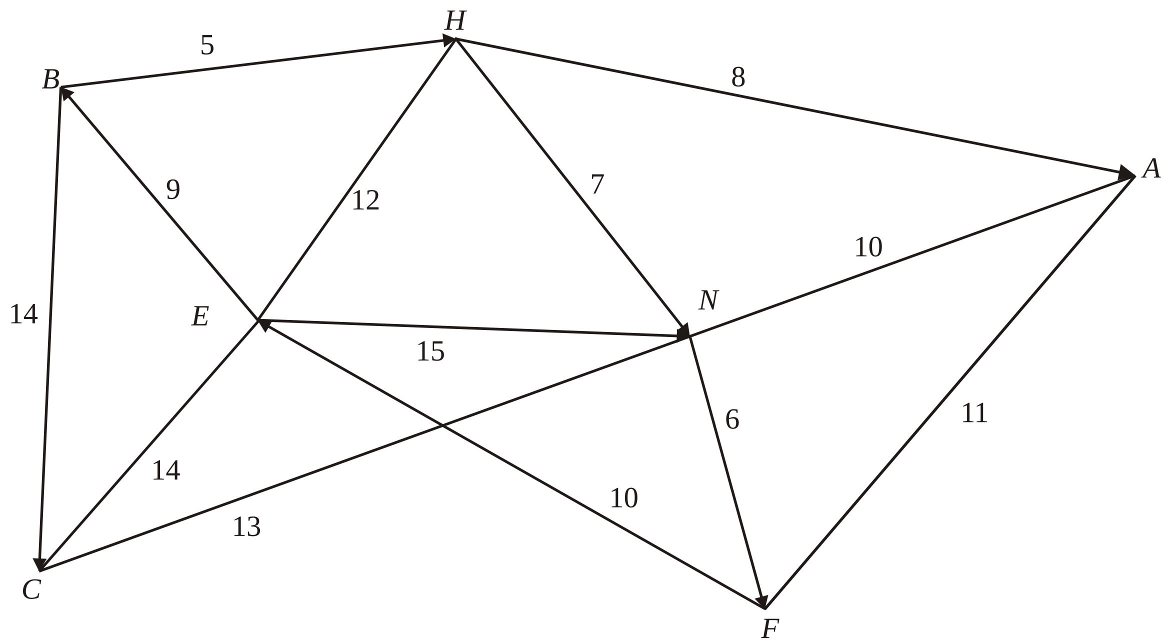
(iii) minimum spanning tree.

(3)

(b) State one difference between Kruskal's algorithm and Prim's algorithm, to find a minimum spanning tree.

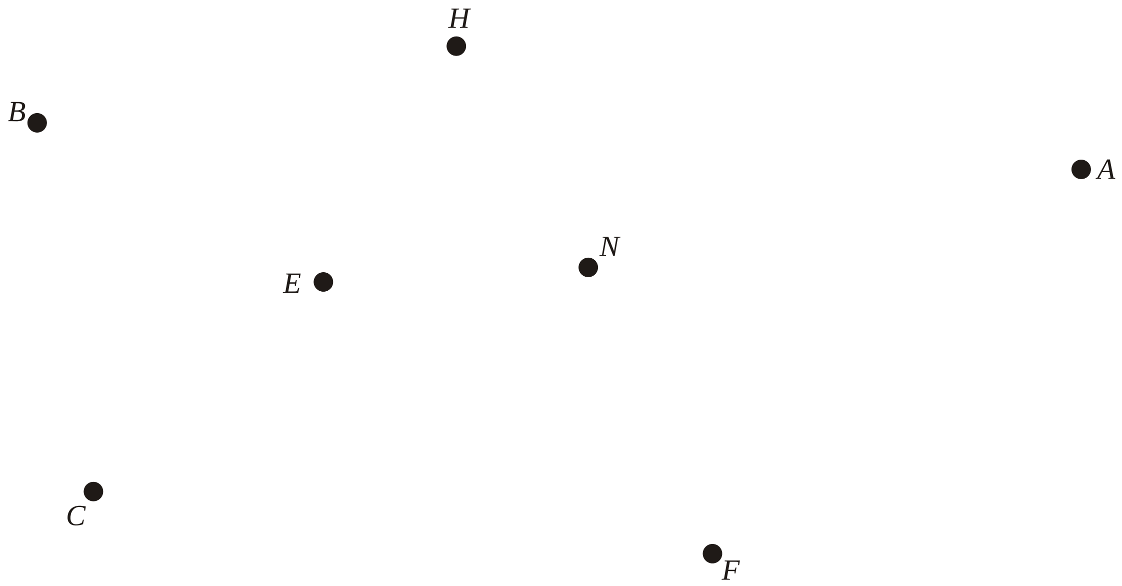
(1)

Figure 1



(c) Use Kruskal's algorithm to find the minimum spanning tree for the network shown in Fig. 1. State the order in which you included the arcs. Draw the minimum spanning tree in Diagram 1 below and state its length.

Diagram 1



Order arcs included……………………………………………………………………………….

Length of minimum spanning tree……………………………………………………………...

(4)

Figure 2

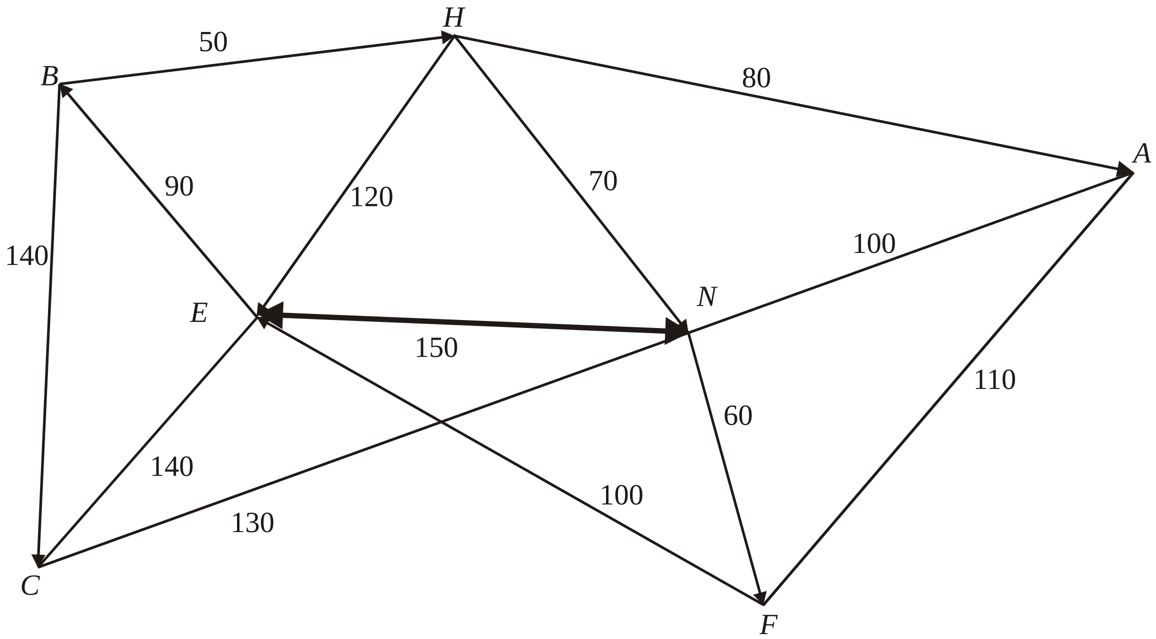
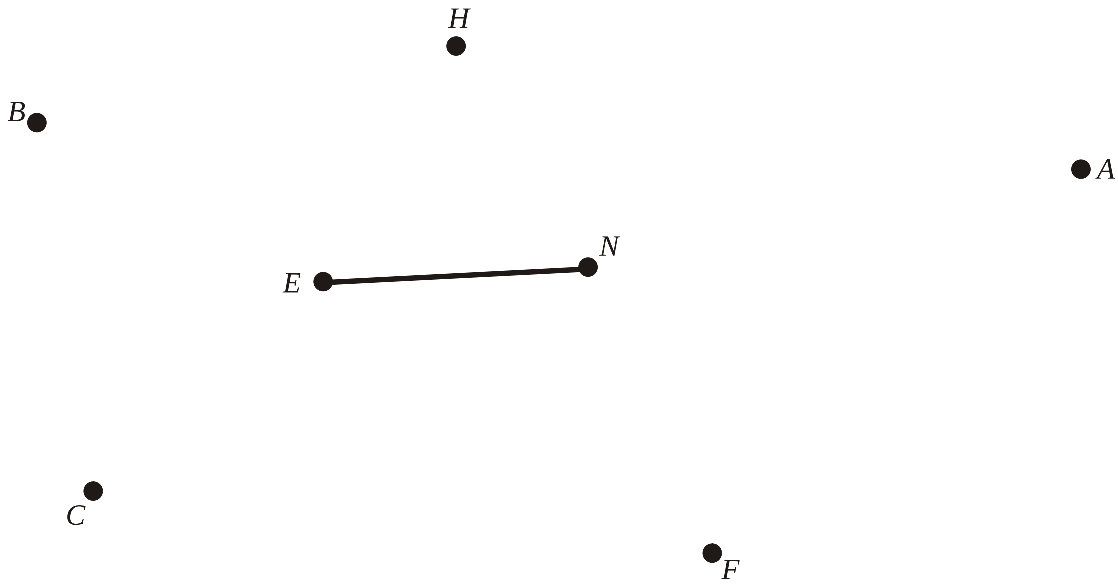

H
B
A
E
N
C
F
50
80
90
120
70
140
100
150
110
60
140
100
130

Figure 2 models a car park. Currently there are two pay-stations, one at *E* and one at *N*. These two are linked by a cable as shown. New pay-stations are to be installed at *B*, *H*, *A*, *F* and *C*. The number on each arc represents the distance between the pay-stations in metres. All of the pay-stations need to be connected by cables, either directly or indirectly. The current cable between *E* and *N* must be included in the final network. The minimum amount of new cable is to be used.

(d) Using your answer to part (*c*), or otherwise, determine the minimum amount of **new** cable needed. Use Diagram 2 to show where these cables should be installed. State the minimum amount of new cable needed.

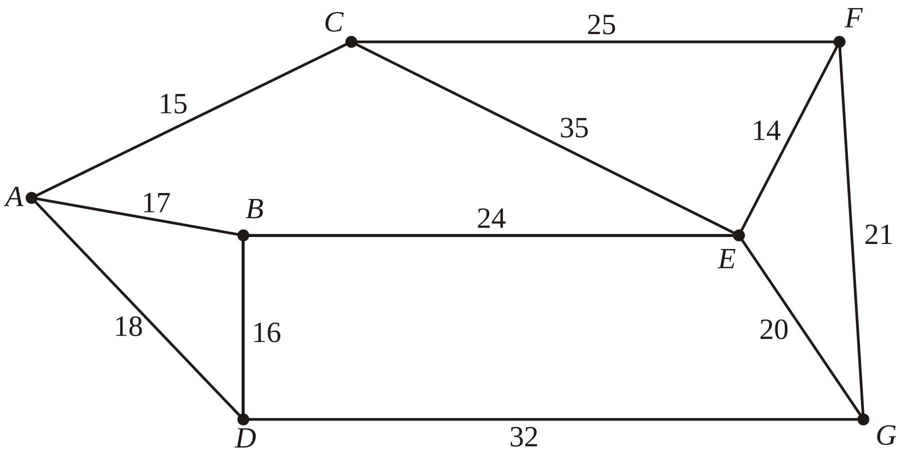
Diagram 2



(3)
(Total 11 marks)

11. (a) Describe the differences between Prim’s algorithm and Kruskal’s algorithm for finding a minimum connector of a network.

(3)



(b) Listing the **arcs** in the order that you select them, find a minimum connector for the network in the diagram above, using

(i) Prim’s algorithm,

(ii) Kruskal’s algorithm.

(4)
(Total 7 marks)

1. (i) FH, AD, DE, CE, (not DC), $\begin{pmatrix} BC \\ EG \end{pmatrix}$, (not AC), CF, HI, (not FI), IJ M1A1A1 3

(ii) AD, DE, EC, $\begin{pmatrix} BC \\ EG \end{pmatrix}$, CF, FH, HI, IJ stop M1A1A1 3

(b) Start off the tree with AB and FI, then apply Kruskal M1 A1 2

[8]

2. (a) DE GF DC $\begin{Bmatrix} \text{not CE} \\ \text{BD} \end{Bmatrix}$ EG (not EF not CF) AC (not AB) GH M1 A1 A1 3

Note

1M1: Kruskal's algorithm – first 4 arcs selected chosen correctly.
1A1: All seven non-rejected arcs chosen correctly.
2A1: All rejections correct and in correct order and at correct time.

(b)

	A	B	C	D	E	F	G	H
A	-	31	30	-	-	-	-	-
B	31	-	-	**24**	-	-	-	38
C	30	-	-	**22**	24	**29**	-	-
D	-	**24**	**22**	-	18	-	-	**34**
E	-	-	24	18	-	28	**26**	-
F	-	-	**29**	-	28	-	21	-
G	-	-	-	-	**26**	21	-	33
H	-	38	-	**34**	-	-	**33**	-

B2, 1, 0 2

Note

1B1: condone two (double) errors
2B1: cao

(c) AC CD DE BD GE GF GH M1 A1 A1 3

Note

1M1: Prim's algorithm – first four arcs chosen correctly, in order, **or** first five nodes chosen correctly, in order.{A,C,D,E,B….}
1A1: First six arcs chosen correctly **or** all 8 nodes chosen correctly, in order. {A,C,D,E,B,G,F,H}
2A1: All correct and arcs chosen in correct order.

(d)	Weight: 174	B1	1

Note

1B1: cao

Starting at	Minimum arcs required for M1	Nodes	order
A	AC CD DE DB	ACDEB(GFH)	15234(768)
B	BD DE DC	BDEC(GFAH)	(7)1423(658)
C	CD DE DB	CDEB(GFAH)	(7)4123(658)
D	DE DC DB	DECB(GFAH)	(7)4312(658)
E	ED DC DB	EDCB(GFAH)	(7)4321(658)
F	FG GE ED DC DB	FGEDCB(AH)	(7)654312(8)
G	GF GE ED DC DB	GFEDCB(AH)	(7)654321(8)
H	HG GF GE	HGFE(DCBA)	(8765)4321

[9]

3. (a) (i) **All pairs** of vertices connected by a **path**, but not describing complete graph. B1

(ii) No cycles B1

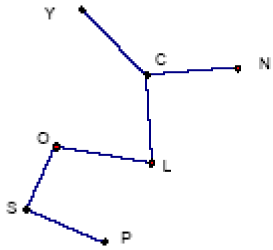
(iii) All nodes connected (accept definition of **minimum** spanning tree) B1 3

(b) Kruskal's (algorithm) B1 1

(c) (i)

L-O 56	Using Prim. first 2 correct	M1	
L-C 60			
C-N 62			
O-S 63	Next 2	A1	
S-P 43			
C-Y 156	Finish	A1	
Total length 440 (miles)	Total	A1 =B1	

(ii) Tree correct



B1 5

Note

Accept weights as indicating arcs.

Misreads – award M1 A0 A0 for these:

- Vertices, not edges given L O C N S P Y
- Numbers across top, edges either incorrect or not given: 3 1 4 2 6 5 7.

Also accept these, misreading And not starting at L – again M1A0A0

Started at	Minimum arcs	nodes	Numbers
C	CL,LO,CN,…..	CLONSPY	1243657
N	NC,CL,LO,OS,SP,CY	NCLOSPY	2314657
O	OL,LC,CN,OS,….	OLCNSPY	3241657
P	PS,SO,OL,LC,CN.CY	PSOLCNY	5463127
S	SP.SO,…	SPOLCNY	5463217
Y	YC,CL,LO,CN,..	YCLONSP	2354761

[9]

4. (a) AD, AE, DB; DC, CF M1 A1; A1 3

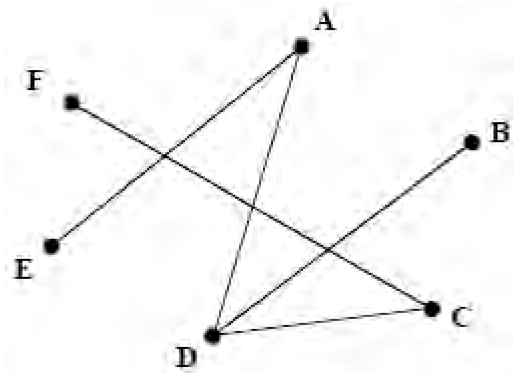
Note

1M1: Using Prim – first 2 arcs probably but condone starting from another vertex.

1A1: first three arcs correct

2A1: all correct.

(b)



B1 1

<u>Note</u>

1B1: CAO

(c) Weight 595 (km) B1 1

<u>Note</u>

1B1: CAO condone lack of km.

<u>Apply the misread rule, if not listing arcs or not starting at A.</u>

So for M1 (only)

Accept numbers across the top (condoning absence of 6)

Accept full vertex listing

Accept full arc listing starting from vertex other than A

[AD AE DB DC CF]	{1 4 5 2 3 6}	ADEBCF
BD AD AE CD CF	{3 1 5 2 4 6}	BDAECF
CD AD AE BD CF	{3 5 1 2 4 6}	CDAEBF
DA AE DB CD CF	{2 4 5 1 3 6}	DAEBCF
EA AD DB DC CF	{2 4 5 3 1 6}	EADBCF
FC CD AD AE BD	{4 6 2 3 5 1}	FCDAEB

[5]

5. (a) e.g.

- Prims starts with any vertex, Kruskal starts with the shortest arc.
- It is not necessary to check for cycles when using Prim.
- Prims adds nodes to the growth tree, Kruskal adds arcs.
- The tree ‘grows’ in a connected fashion when using Prim.
- Prim can be used when data in a matrix form.

Other correct statements also get credit. B2,1,0 2

1B1: Generous one coprrect difference. If bod give B1

2B1: Generous two distinct, correct differences.

(b) (i) e.g. AC, CF, FD, DE, DG, AB. M1, A1, A1 3

(ii) CF, DE, DF, not CD, not EF, DG, not FG, not EG, AC, not AD, AB.
[18, 19, 20, not 21, not 21, 22, not 23, not 24, 25, not 26, 27]
M1, A1, A1 3

1M1: Prim's algorithm – first three arcs chosen correctly, in order, or first four nodes chosen correctly, in order.

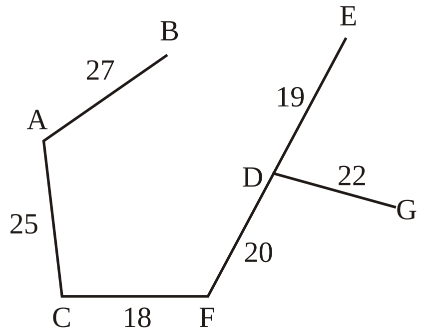
1A1: First five arcs chosen correctly; all 7 nodes chosen correctly, in order.

2A1: All correct and arcs chosen in correct order.

2M1: Kruskal's algorithm – first 4 arcs selected chosen correctly.

1A1: All six non-rejected arcs chosen correctly.

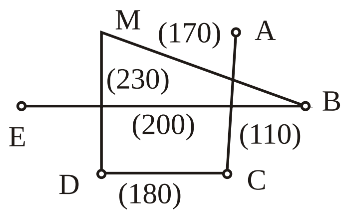
2A1: All regions correct and in correct order and ta correct time.



[8]

6. (a) *MB, BE, MD, DC, CA* M1A1A1 3

(b)



B1ft 1

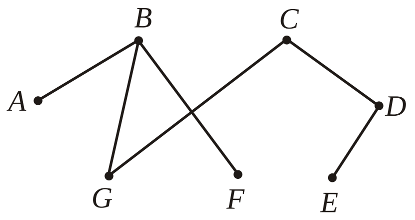
(c) 170 + 200 + 210 + 180 + 100 = 860 B1 1

(d) (A cycle is formed when an arc is used that connects two vertices already connected to each other in the tree) B2,1,0 2
Prim's algorithm always selects arcs that bring a vertex not in the tree into the tree, so cycles can't happen

[7]

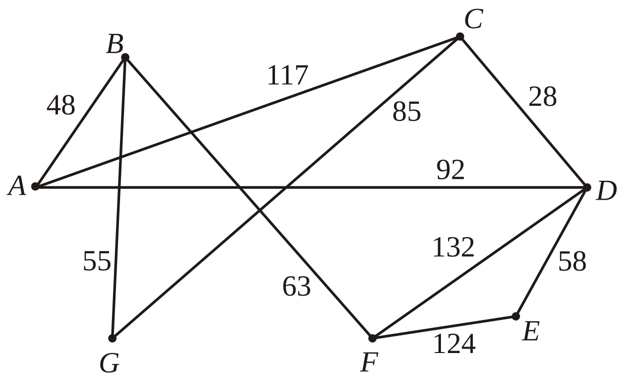
7. (a) AB, BG, BF | GC, CD, DE {1 2 5 6 7 4 3} M1 A1 A1 3

weight 337 m B1



B1ft 2

(b)



M1 A1 A1 3

(c) AB + CF = 48 + 160 = 208 M1 A1

AC + BF = 117 + 63 = 180 A1

AF + B C = 111 + 140 = 251 A1 4

e.g. A B F B G C A C D E F D A A1

length 802 + 180 = 982 m M1 A1ft 3

[15]

8. (a) (i) FH,AD,DE,CE, (not DC), $\begin{Bmatrix} BC \\ EG \end{Bmatrix}$, (not AC), CF, HI,

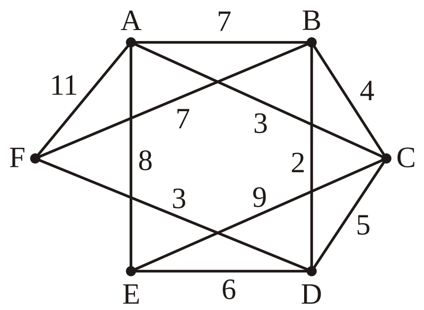
(not FI), IJ stop M1 A1 A1 3

(ii) AD, DE, EC, $\begin{Bmatrix} BC \\ EG \end{Bmatrix}$, CF, FH, HI, IJ stop M1 A1 A1 3

(b) Start off the tree with AB and FI, then apply Kruskal B2, 1, 0 2

[8]

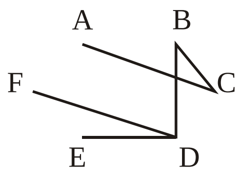
9. (a)



B1

B1 2

(b)



M1

BD, $\left(\frac{A\ C}{D\ F}\right)$, BC, Not CD, DE A1, A1

B1

Length = 18 km B1 5

(c) DB, DF, BC, CA, DE [5,2,4,1,6,3,] M1 A1 A1 3

[10]

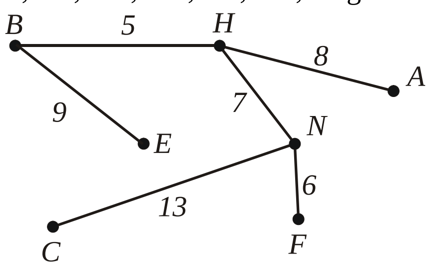
10. (a) (i) A connected graph with no cycles, loops or multiple edges B1

(ii) A tree that includes all vertices B1

(iii) A spanning tree of minimum total length B1 3

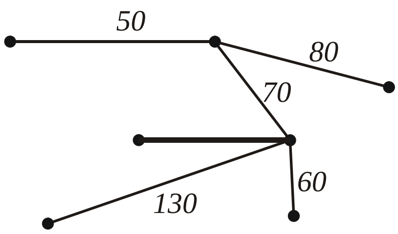
(b) E.g.
- In Kruskal the shortest arc is added (unless it completes a cycle), in Prim the nearest unattached vertex is added
- There is no need to check for cycles when using Prim, but there is when using Kruskal — B1 — 1
- In Prim the tree always "grows" in a connected fashion
- Kruskal starts with the shortest edge, Prim with any vertex

(c) BH, NF, HN, HA, BE, NC; length = 48 — M1 A1; B1



B1 — 4

(d)



B1

New cable – 390m — M1 A1 — 3

[11]

11. (a) e.g.:
- In Prim the tree always 'grows' in a connected fashion; — B3,2,1,0 — 3
- In Kruskal the shortest arc is added (unless it completes a cycle), in Prim the nearest unattached vertex is added;
- There is no need to check for cycles when using Prim;
- Prim can be easily used when network given in matrix form
- Prim stats with a vertex, Kruskal with an edge

(b) (i) Either *AC*, *AB*, *BD*, *BE*, *EF*, *EG* (if starts at *A* or *C*)
or *BD*, *BA*, *AC*, *BE*, *EF*, *EG* (if starts at *B* or *D*)
or *EF*, *EG*, *BE*, *BD*, *BA*, *AC* (if starts at *E* or *F*)
or *GE*, *EF*, *BE*, *BD*, *BA*, *AC* (if starts at *G*) — M1 A1

(ii) *EF*, *AC*, *BD*, *BA*, *EG*, *BE* — M1 A1 — 4

[7]

1. No Report available for this question.

2. Around 45% of the candidates gained full marks on this question. Candidates were directed to list the **arcs** in the order in which they included them in the tree, but many candidates did not do so.

In part (a) a number of candidates only stated the arcs they were including in their tree and did not state the arcs that they rejected, as they rejected them. Some candidates only referred to the length of the arc rather than by its end vertices, this makes it difficult for the examiners to determine which arc is being considered.

Part (b) was usually completed correctly.

In part (c) many candidates showed their working on the table but then did not list the arcs in the correct order, often adding BD too late.

Many candidates completed part (d) correctly.

3. The definitions in part (a) challenged many, particularly that of a 'connected graph' which challenged even the better candidates, with some defining a complete graph and others being unclear about the nature of the connection between nodes. There were the usual confusions of technical terms, 'nodes connected by vertices' etc. Part (b) was often correct, although often misspelled, but Prim and Dijkstra were also popular incorrect answers. Most candidates were able to make progress with part (c), but many lost marks because they did not list the **edges** in order of selection, or did not start at L as demanded in the question. The correct tree and its weight were often seen.

4. This was a good starter and was well answered. Some candidates headed the columns correctly but then did not **list** the arcs correctly; they should be encouraged to list each arc as it is selected. As expected the commonest error was to look for the smallest entry in the latest column rather than in all numbered columns.

5. Part (a) caused problems for some candidates. Candidates were asked for differences between the two algorithms and many simply made statements such as Prim uses nodes without going on to say the Kruskal uses edges. A popular answer was to mention ordering the arcs into ascending order for Kruskal; if this used as a difference candidates must indicate why this is important when using the algorithm. There was the usual muddled use of basic technical terminology. Some candidates muddled the two algorithms. Part (b) as usually well-answered although some just drew their MST and did not list the arcs, in order, as instructed. Some wasted time in Prim by converting to the matrix form. Some did not make their rejections clear when applying Kruskal's algorithm.

6. There were many fully correct answers to (a) seen, but also many who applied the nearest neighbour algorithm instead. Many seemed to confuse arcs with vertices and did not list their arcs as required. Most candidates drew a tree in part (b) and almost all who got the correct tree also correctly found its weight. Many candidates were able to describe the mechanics of the algorithm in part (d), with the commonly seen reference to 'crossing out rows', but only the best were able to think about HOW the algorithm worked and secure both marks.

7. This proved an accessible question, but careless slips resulted in few gaining full credit. Not all candidates used Prim's algorithm correctly and of those that did, some did not clearly state the order in which the edge were selected – as directed in the question. A disappointing number of candidates used the nearest neighbour algorithm instead of Prim's algorithm. Many omitted to state the weight of the tree. The vast majority were able to draw the network in part (b). The majority did three pairings but many did not find all the shortest routes between the pairs. Some discarded any route involving more than one edge and some did not state the totals of their pairings. A few did not state a route and some found a semi-Eulerian solution.

8. There was a varied response to this question, with some very good and some quite poor responses seen. Part (a) was generally better attempted than part (b). In part (a) the candidates were asked to make the order in which they selected the arcs clear, many did not do this. Many candidates wasted time by drawing a succession of diagrams showing the addition of one arc at a time. Many candidates lost marks in Kruskal's algorithm by not showing the rejection of the arcs that created loops. In some attempts at Prim's algorithm, many did not list the arcs and others referred to rejecting arcs to prevent cycles from forming. Many wasted time in drawing a matrix for Prim. Part (b) was often poorly done with many candidates opting for Prim despite the two arcs not being connected. Of those who correctly selected Kruskal, only a few were able to give a coherent explanation.

9. Almost all of the candidates were able to draw the arcs correctly but it was not always clear which numbers went with which arcs, which lead to mistakes later. Most candidates were able to perform Kruskal’s algorithm correctly although a number omitted to state that arc CD was rejected. Most went on to correctly draw the minimum spanning tree, but units were often omitted when stating its weight. Prim’s algorithm was less well done. Some did not show that they had used the matrix, some started at A and many listed vertices instead of arcs.

10. The majority of candidates found part (a) tricky. There were some very poor definitions, poor use of technical terms and muddled thinking seen. Candidates often did not write in complete sentences, making it very difficult to understand the point they were trying to make. Part (b) was usually better done although candidates confused arcs, vertices, nodes and edges and made some worrying remarks about cycles. Part (c) was usually completed correctly. Part (d) caused problems for some candidates, with some incorrect diagrams and incorrect answers of 54m, 540m and 39m often seen.

11. Very few candidates attempted a comparison in part (a). Most just described the two methods. A significant number confused the two algorithms. There was some poor use of technical terms seen. Part (b) was very well done with most candidates gaining full marks, but a few did not list the arcs as requested. A surprising number of candidates seem to think that Prim’s algorithm can only be applied to a matrix and wasted a lot of time in creating the matrix. Others wasted time (and often went onto at least two extra sides of paper) by drawing a ‘cartoon sequence’ showing the state of the tree after each arc addition, rather than simply listing the arcs in order.