D1 Revision Questions

Using questions from June 2001, January 2002, June 2002

Critical Path Analysis

4 Critical path analysis

What students need to learn:

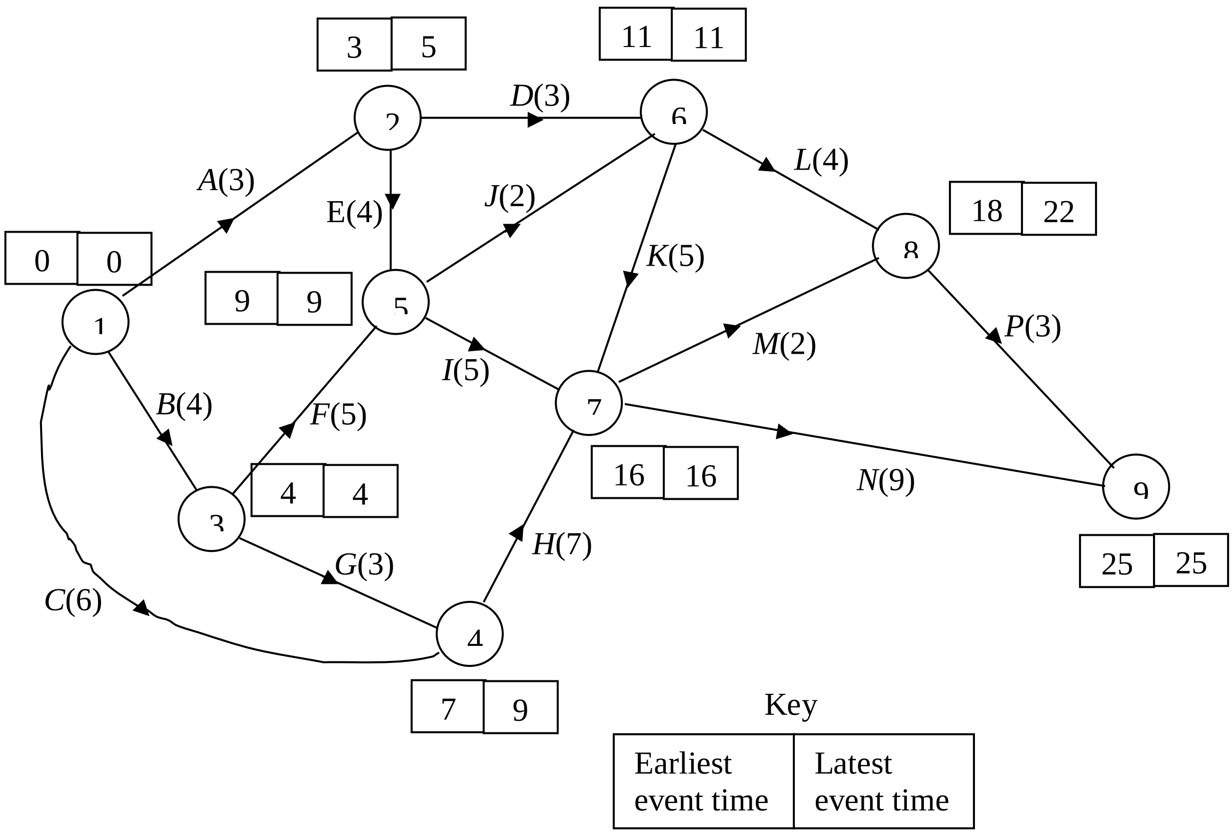
Modelling of a project by an activity network, from a precedence table.	Activity on arc will be used. The use of dummies is included. In a precedence network, precedence tables will only show immediate predecessors.
Completion of the precedence table for a given activity network.	
Algorithm for finding the critical path. Earliest and latest event times. Earliest and latest start and finish times for activities.	
Total float. Gantt (cascade) charts. Scheduling.	

1. The precedence table for activities involved in a small project is shown below

Activity	Preceding Activities
A	–
B	–
C	–
D	*B*
E	*A*
F	*A*
G	*B*
H	*C*, *D*
I	*E*
J	*E*
K	*F*, *G*, *I*
L	*H*, *J*, *K*

Draw an activity network, using activity on edge and without using dummies, to model this project. **(5)**

2. **Figure 4**



A building project is modelled by the activity network shown in Fig. 4. The activities are represented by the arcs. The number in brackets on each arc gives the time, in hours, taken to complete the activity. The left box entry at each vertex is the earliest event time and the right box entry is the latest event time.

(*a*) Determine the critical activities and state the length of the critical path. **(2)**

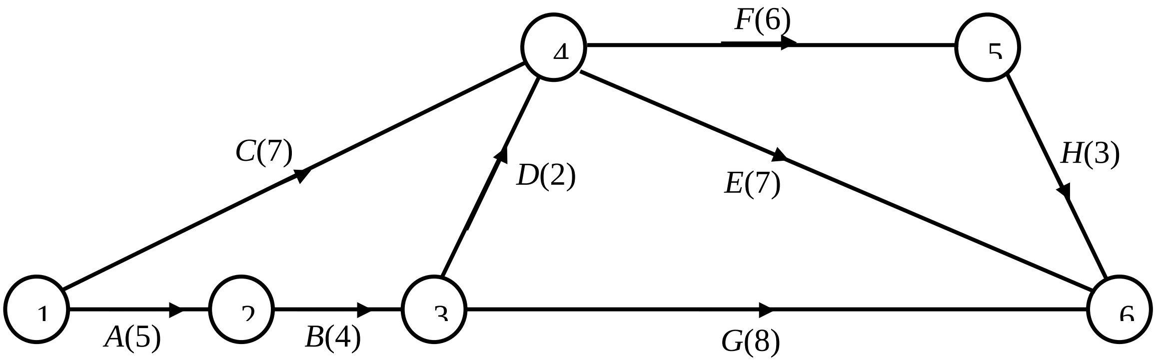
(*b*) State the total float for each non-critical activity. **(3)**

(*c*) On the grid in the answer booklet, draw a cascade (Gantt) chart for the project. **(4)**

Given that each activity requires one worker,

d) draw up a schedule to determine the minimum number of workers required to complete the project in the critical time. State the minimum number of workers. **(3)**

3. **Figure 3**



A project is modelled by the activity network shown in Fig 3. The activities are represented by the edges. The number in brackets on each edge gives the time, in days, taken to complete the activity.

(a) Calculate the early time and the late time for each event. Write these in the boxes on the answer sheet. **(4)**

(*b*) Hence determine the critical activities and the length of the critical path. **(2)**

(*c*) Obtain the total float for each of the non-critical activities. **(3)**

(*d*) On the first grid on the answer sheet, draw a cascade (Gantt) chart showing the information obtained in parts (*b*) and (*c*). **(4)**

Each activity requires one worker. Only two workers are available.

(*e*) On the second grid on the answer sheet, draw up a schedule and find the minimum time in which the 2 workers can complete the project. **(4)**

Algorithms

2 Algorithms on graphs

What students need to learn:

The minimum spanning tree (minimum connector) problem. Prim's and Kruskal's (greedy) algorithm.	Matrix representation for Prim's algorithm is expected. Drawing a network from a given matrix and writing down the matrix associated with a network will be involved.
Dijkstra's algorithm for finding the shortest path.	

4. **Figure 1**

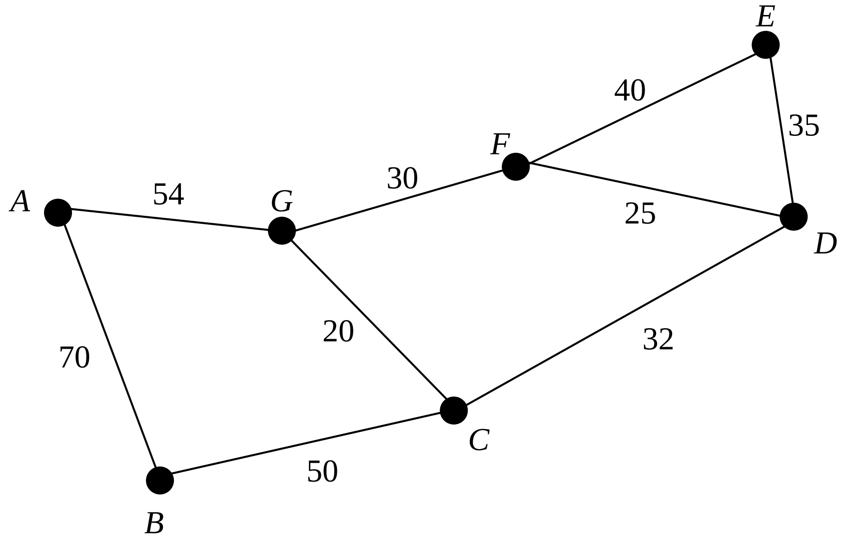


Figure 1 shows 7 locations *A*, *B*, *C*, *D*, *E*, *F* and *G* which are to be connected by pipelines. The arcs show the possible routes. The number on each arc gives the cost, in thousands of pounds, of laying that particular section.

(*a*) Use Kruskal's algorithm to obtain a minimum spanning tree for the network, giving the order in which you selected the arcs. **(4)**

(*b*) Draw your minimum spanning tree and find the least cost of the pipelines. **(3)**

5. (i)

	A	*B*	*C*	*D*	*E*	*F*
A	–	10	12	13	20	9
B	10	–	7	15	11	7
C	12	7	–	11	18	3
D	13	15	11	–	27	8
E	20	11	18	27	–	18
F	9	7	3	8	18	–

The table shows the distances, in metres, between six nodes *A*, *B*, *C*, *D*, *E*, and *F* of a network.

(*a*) Use Prim's algorithm, starting at *A*, to solve the minimum connector problem for this table of distances. Explain your method and indicate the order in which you selected the edges. **(4)**

(*b*) Draw your minimum spanning tree and find its total length. **(2)**

(*c*) State whether your minimum spanning tree is unique. Justify your answer. **(1)**

(ii) A connected network *N* has seven vertices.

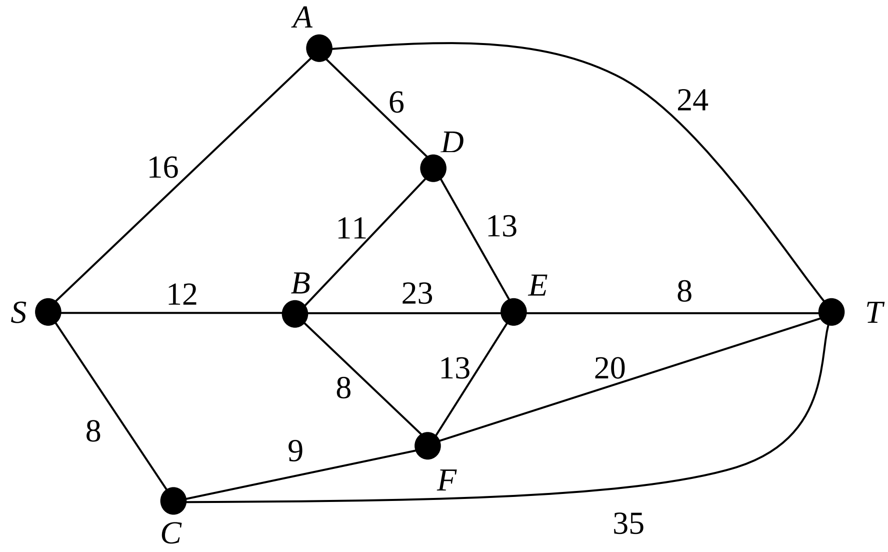
(*a*) State the number of edges in a minimum spanning tree for *N*. **(1)**

A minimum spanning tree for a connected network has *n* edges.

(*b*) State the number of vertices in the network. **(1)**

6.

Figure 3



The weighted network shown in Fig. 3 models the area in which Bill lives. Each vertex represents a town. The edges represent the roads between the towns. The weights are the lengths, in km, of the roads.

(*a*) Use Dijkstra's algorithm to find the shortest route from Bill's home at *S* to *T*. Complete all the boxes on the answer sheet and explain clearly how you determined the path of least weight from your labelling. **(8)**

Bill decides that on the way to T, he must visit a shop in town *E*.

(*b*) Obtain his shortest route now, giving its length and explaining your method clearly**(3)**

7.

Figure 1

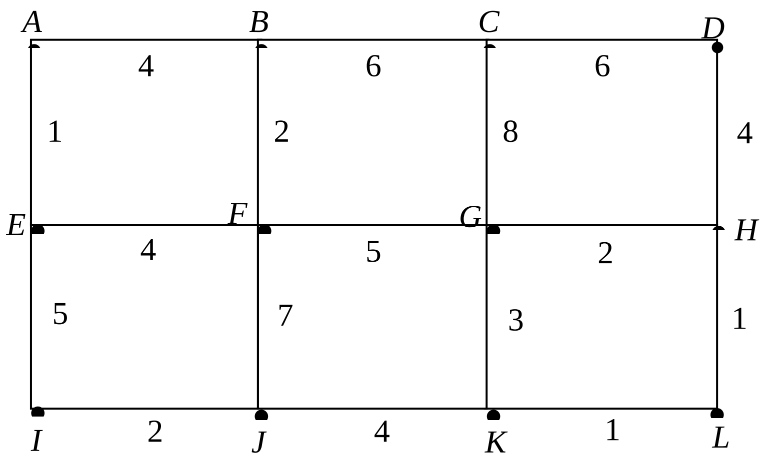
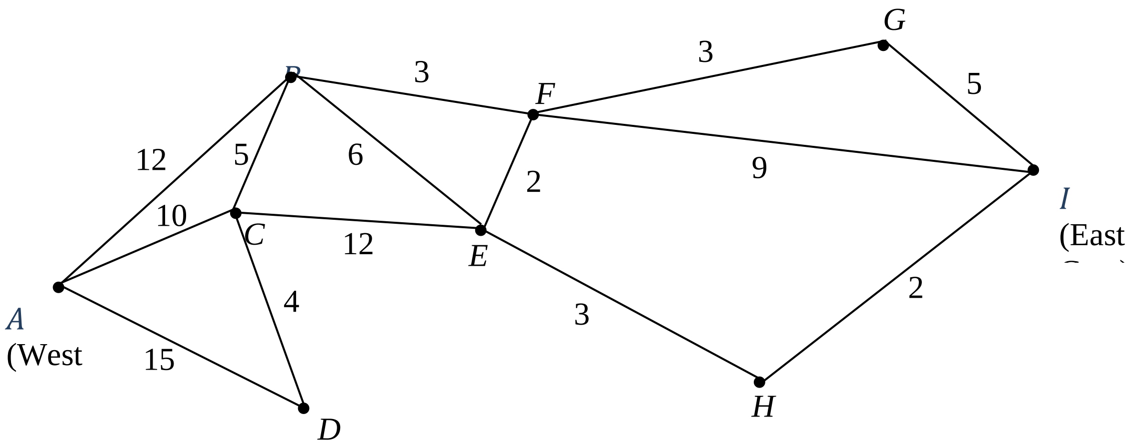


Figure 1 shows a network of roads. Erica wishes to travel from *A* to *L* as quickly as possible. The number on each edge gives the time, in minutes, to travel along that road.

(*a*) Use Dijkstra's algorithm to find a quickest route from *A* to *L*. Complete all the boxes on the answer sheet and explain clearly how you determined the quickest route from your labelling. **(7)**

(*b*) Show that there is another route which also takes the minimum time **(1)**



8. **Figure 3**

Figure 3 shows the network of paths in a country park. The number on each path gives its length in km. The vertices *A* and *I* represent the two gates in the park and the vertices *B*, *C*, *D*, *E*, *F*, *G* and *H* represent places of interest.

(*a*) Use Dijkstra's algorithm to find the shortest route from *A* to *I*. Show all necessary working in the boxes in the answer booklet and state your shortest route and its length.

(5)

The park warden wishes to check each of the paths to check for frost damage. She has to cycle along each path at least once, starting and finishing at *A*.

(*b*) (i) Use an appropriate algorithm to find which paths will be covered twice and state these paths.

(ii) Find a route of minimum length.

(iii) Find the total length of this shortest route.

(5)

Route Inspection

3 The route Inspection problem

What students need to learn:

Algorithm for finding the shortest route around a network, travelling along every edge at least once and ending at the start vertex. The network will have up to four odd nodes.	Also known as the 'Chinese postman' problem. Students will be expected to use inspection to consider all possible pairings of odd nodes.

9. **Figure 2**

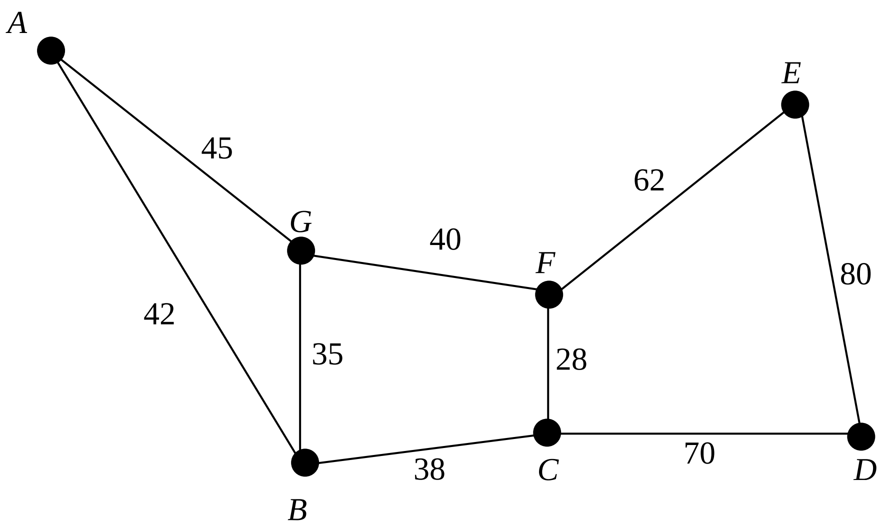


Figure 2 shows a new small business park. The vertices *A*, *B*, *C*, *D*, *E*, *F* and *G* represent the various buildings and the arcs represent footpaths. The number on an arc gives the length, in metres, of the path. The management wishes to inspect each path to make sure it is fit for use.

Starting and finishing at *A*, solve the Route Inspection (Chinese Postman) problem for the network shown in Fig. 2 and hence determine the minimum distance that needs to be walked in carrying out this inspection. Make your method and working clear and give a possible route of minimum length.

(7)

10. **Figure 2**

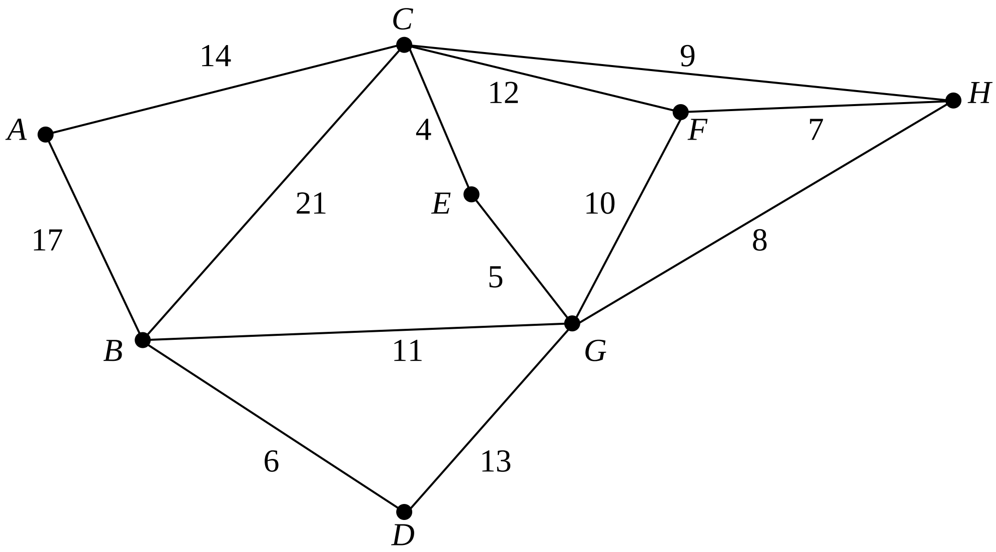


Figure 2 models an underground network of pipes that must be inspected for leaks. The nodes *A*, *B*, *C*, *D*, *E*, *F*, *G* and *H* represent entry points to the network. The number on each arc gives the length, in metres, of the corresponding pipe.

Each pipe must be traversed at least once and the length of the inspection route must be minimised.

(*a*) Use the Route Inspection algorithm to find which paths, if any, need to be traversed twice.

(4)

It is decided to start the inspection at node *C*. The inspection must still traverse each pipe at least once but may finish at any node.

(*b*) Explaining your reasoning briefly, determine the node at which the inspection should finish if the route is to be minimised. State the length of your route.

(3)

Algorithms

1 Algorithms

What students need to learn:

The general ideas of algorithms and the implementation of an algorithm given by a flow chart or text.	The order of an algorithm is not expected. Whenever finding the middle item of any list, the method defined in the glossary must be used.
Students should be familiar with bin packing, bubble sort, quick sort, binary search.	When using the quick sort algorithm, the pivot should be chosen as the middle item of the list.

11. 90, 50, 55, 40, 20, 35, 30, 25, 45

(*a*) Use the bubble sort algorithm to sort the list of numbers above into descending order showing the rearranged order after each pass.

(5)

Jessica wants to record a number of television programmes onto video tapes. Each tape is 2 hours long. The lengths, in minutes, of the programmes she wishes to record are:

55, 45, 20, 30, 30, 40, 20, 90, 25, 50, 35 and 35.

(*b*) Find the total length of programmes to be recorded and hence determine a lower bound for the number of tapes required.

(2)

(*c*) Use the first fit decreasing algorithm to fit the programmes onto her 2-hour tapes.

(3)

Jessica's friend Amy says she can fit all the programmes onto 4 tapes.

(*d*) Show how this is possible.

(2)

12. (i) Use the binary search algorithm to try to locate the name *SABINE* in the following alphabetical list. Explain each step of the algorithm.

1. *ABLE*
2. *BROWN*
3. *COOKE*
4. *DANIEL*
5. *DOUBLE*
6. *FEW*
7. *OSBORNE*
8. *PAUL*
9. *SWIFT*
10. *TURNER*

(5)

(ii) Find the maximum number of iterations of the binary search algorithm needed to locate a name in a list of 1000 names.

(2)

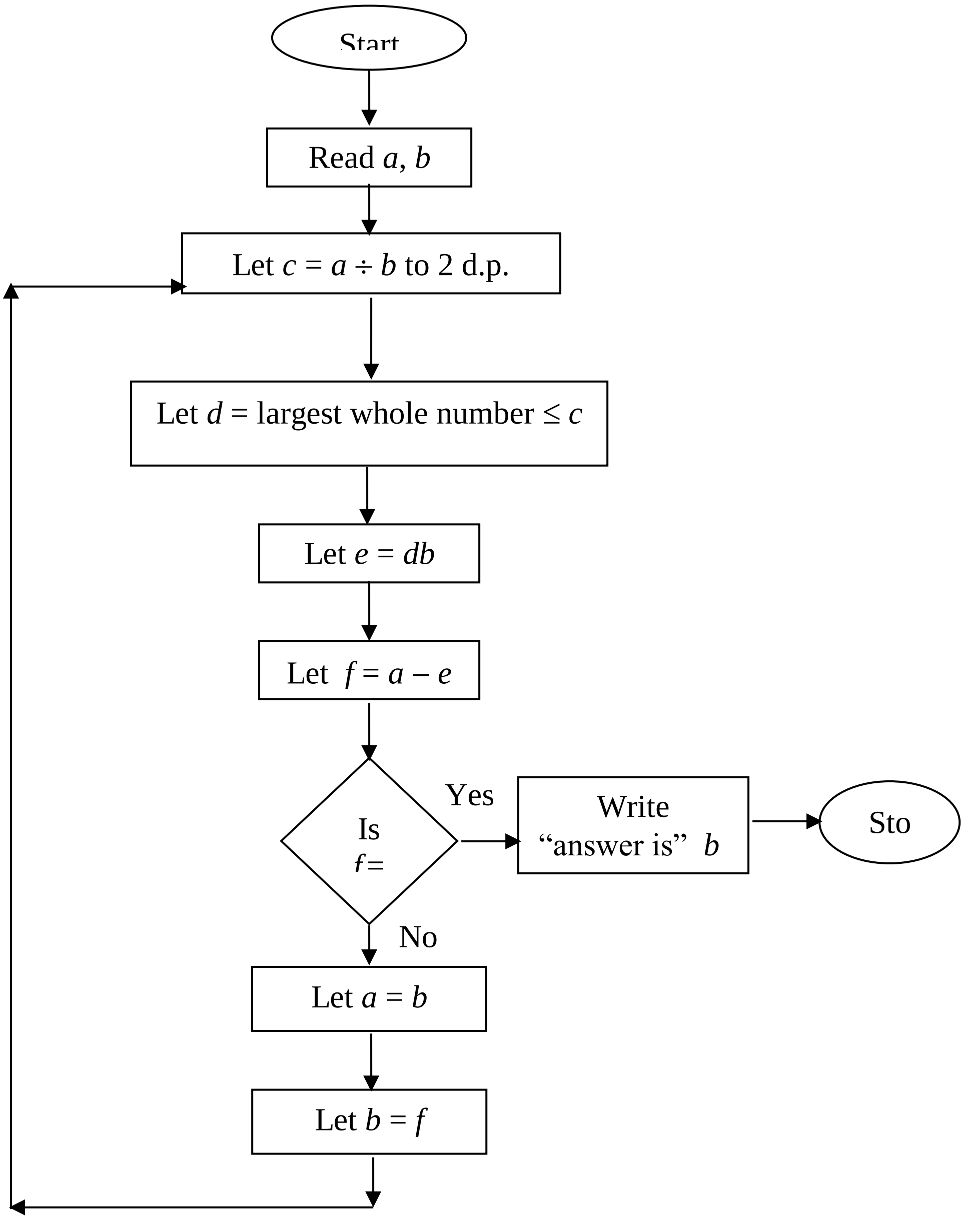
13.

Ashford	6
Colnbrook	1
Datchet	18
Feltham	12
Halliford	9
Laleham	0
Poyle	5
Staines	13
Wraysbury	14

The table above shows the points obtained by each of the teams in a football league after they had each played 6 games. The teams are listed in alphabetical order. Carry out a quick sort to produce a list of teams in descending order of points obtained.

(5)

14. An algorithm is described by the flow chart below.



(*a*) Given that $a = 645$ and $b = 255$, complete the table in the answer booklet to show the results obtained at each step when the algorithm is applied.

(7)

(*b*) Explain how your solution to part (*a*) would be different if you had been given that $a = 255$ and $b = 645$.

(3)

(*c*) State what the algorithm achieves. **(1)**

Matchings

6 Matchings

What students need to learn:

Use of bipartite graphs for modelling matchings. Complete matchings and maximal matchings.	Students will be required to use the maximum matching algorithm to improve a matching by finding alternating paths. No consideration of assignment is required.
Algorithm for obtaining a maximum matching.	

15. Ann, Bryn, Daljit, Gareth and Nickos have all joined a new committee. Each of them is to be allocated to one of five jobs 1, 2, 3, 4 or 5. The table shows each member's preferences for the jobs.

Ann	1 or 2
Bryn	3 or 1
Daljit	2 or 4
Gareth	5 or 3
Nickos	1 or 2

Initially Ann, Bryn, Daljit and Gareth are allocated the first job in their lists shown in the table.

(*a*) Draw a bipartite graph to model the preferences shown in the table and indicate, in a distinctive way, the initial allocation of jobs.

(2)

(*b*) Use the matching improvement algorithm to find a complete matching, showing clearly your alternating path.

(3)

(*c*) Find a second alternating path from the initial allocation.

(1)

16.

Figure 1

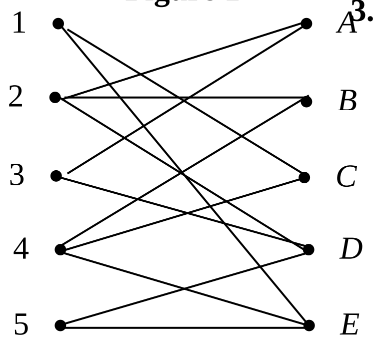
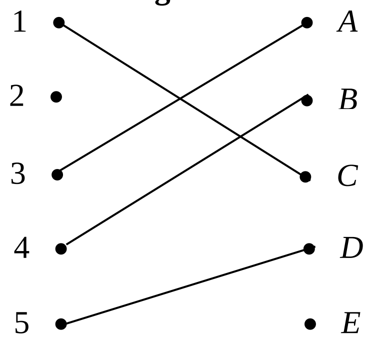


Figure 2



Five members of staff 1, 2, 3, 4 and 5 are to be matched to five jobs *A*, *B*, *C*, *D* and *E*. A bipartite graph showing the possible matchings is given in Fig. 1 and an initial matching *M* is given in Fig. 2.

There are several distinct alternating paths that can be generated from *M*. Two such paths are

2 — *B* = 4 — *E*

and 2 — *A* = 3 — *D* = 5 — *E*

(*a*) Use each of these two alternating paths, in turn, to write down the complete matchings they generate.

(2)

Using the maximum matching algorithm and the initial matching *M*,

(*b*) find two further distinct alternating paths, making your reasoning clear.

(4)

Linear programming

5 Linear programming

What students need to learn:

Formulation of problems as linear programs.

Graphical solution of two variable problems using ruler and vertex methods

Consideration of problems where solutions must have integer values.

17. Two fertilizers are available, a liquid X and a powder Y. A bottle of X contains 5 units of chemical A, 2 units of chemical B and $\frac{1}{2}$ unit of chemical C. A packet of Y contains 1 unit of A, 2 units of B and 2 units of C. A professional gardener makes her own fertilizer. She requires at least 10 units of A, at least 12 units of B and at least 6 units of C.
She buys x bottles of X and y packets of Y.
(*a*) Write down the inequalities which model this situation. **(4)**
(*b*) On the grid provided construct and label the feasible region. **(3)**
A bottle of X costs £2 and a packet of Y costs £3.
(*c*) Write down an expression, in terms of x and y, for the total cost £T. **(1)**
(*d*) Using your graph, obtain the values of x and y that give the minimum value of T. Make your method clear and calculate the minimum value of T. **(4)**
(*e*) Suggest how the situation might be changed so that it could no longer be represented graphically. **(2)**

18. A chemical company produces two products X and Y. Based on potential demand, the total production each week must be at least 380 gallons. A major customer's weekly order for 125 gallons of Y must be satisfied.
Product X requires 2 hours of processing time for each gallon and product Y requires 4 hours of processing time for each gallon. There are 1200 hours of processing time available each week. Let x be the number of gallons of X produced and y be the number of gallons of Y produced each week.
(*a*) Write down the inequalities that x and y must satisfy. **(3)**
It costs £3 to produce 1 gallon of X and £2 to produce 1 gallon of Y. Given that the total cost of production is £C,
(*b*) express C in terms of x and y. **(1)**
The company wishes to minimise the total cost.
(*c*) Using the graphical method, solve the resulting Linear Programming problem. Find the optimal values of x and y and the resulting total cost. **(7)**
(*d*) Find the maximum cost of production for all possible choices of x and y which satisfy the inequalities you wrote down in part (*a*). **(3)**

Glossary for D1

1 Algorithms

In a list containing N items the 'middle' item has position $[\frac{1}{2}(N+1)]$ if N is odd $[\frac{1}{2}(N+2)]$ if N is even, so that if $N = 9$, the middle item is the 5th and if $N = 6$ it is the 4th.

2 Algorithms on graphs

A **graph** G consists of points (**vertices** or **nodes**) which are connected by lines (**edges** or **arcs**).

A **subgraph** of G is a graph, each of whose vertices belongs to G and each of whose edges belongs to G.

If a graph has a number associated with each edge (usually called its **weight**) then the graph is called a **weighted graph** or **network**.

The **degree** or **valency** of a vertex is the number of edges incident to it. A vertex is **odd** (**even**) if it has **odd** (**even**) degree.

A **path** is a finite sequence of edges, such that the end vertex of one edge in the sequence is the start vertex of the next, and in which no vertex appears more then once.

A **cycle** (**circuit**) is a closed path, ie the end vertex of the last edge is the start vertex of the first edge.

Two vertices are **connected** if there is a path between them. A graph is **connected** if all its vertices are connected.

If the edges of a graph have a direction associated with them they are known as **directed edges** and the graph is known as a **digraph**.

A **tree** is a connected graph with no cycles.

A **spanning tree** of a graph G is a subgraph which includes all the vertices of G and is also a tree.

A **minimum spanning tree** (MST) is a spanning tree such that the total length of its arcs is as small as possible. (MST is sometimes called a **minimum connector**.)

A graph in which each of the n vertices is connected to every other vertex is called a **complete graph**.

4 Critical path analysis

The **total float** $F(i, j)$ of activity (i, j) is defined to be $F(i, j) = l_j - e_i$ – duration (i, j), where e_i is the earliest time for event i and l_j is the latest time for event j.

6 Matchings

A **bipartite graph** consists of two sets of vertices X and Y. The edges only join vertices in X to vertices in Y, not vertices within a set. (If there are r vertices in X and s vertices in Y then this graph is $K_{r,s}$.)

A **matching** is the pairing of some or all of the elements of one set, X, with elements of a second set, Y. If every member of X is paired with a member of Y the matching is said to be a **complete matching**.