

1

Algorithms

This chapter will show you how to

- understand the term algorithm
- follow an algorithm in the form of a list of instructions or a flowchart
- put items into groupings of restricted size using 'bin-packing' algorithms
- put lists into numerical or alphabetical order using sorting algorithms
- search for items in a list using the binary search algorithm.

Introduction

The term *Decision Mathematics* is used to cover a number of techniques for solving large-scale data processing and organisational problems.

Much of the underlying theory has been around for a long time, but its application only became possible with the development of computers. It now has an increasingly vital role to play in the world of industry and commerce.

This chapter will tell you about **algorithms**. Every technique in Decision Mathematics is built around a set of instructions – an algorithm – for solving a problem. You will become familiar with the idea of an algorithm, and learn about some specific examples.

You do not need knowledge of any particular mathematical topics in order to understand this chapter. You do however need to be able to show your working in a detailed, well-organised and logical way.

An **algorithm** is a well-defined sequence of steps leading to the solution of a problem of a given type.

Example 1 should give you a feel for the notion of an algorithm.

EXAMPLE 1

Find a procedure for completing the following simple puzzle. Six counters – three black and three white – are placed on a grid of seven squares, as shown.



The white counters may only move left to right, either sliding into a vacant space or jumping one black counter into a vacant space. The black counters move right to left under the same rules. The aim is to reverse the positions of the two sets of counters.

To state the solution to the puzzle, you could just list all fifteen moves needed. However, it is better to write a list of instructions so that the reader can decide on the right move at each stage.

One possible list is:

- Step 1 Decide at random which colour is the ‘active’ colour
- Step 2 Make the available active colour move
- Step 3 If no further active colour move is possible, go to Step 6
- Step 4 If the next available active colour move is a second slide move, do not make the move but go to Step 6
- Step 5 Make the next available active colour move and go to Step 3
- Step 6 If the puzzle is complete then stop
- Step 7 Change the active colour and go to Step 2

If you are not familiar with the solution to this puzzle, you might like to give it a try before reading on.

You will not be expected to write algorithms from scratch, though you may be required to complete or modify an algorithm.

The instructions in Example 1 are more general than just the solution of the original puzzle. They work for any number of counters of each colour, even if the numbers of white counters and black counters are different. They form an algorithm, enabling anyone to solve all problems of this type.

You might like to try applying these instructions to another problem of this type.

An algorithm should

- enable someone to solve all problems of a particular type, just by following the instructions – no insight should be needed
- provide a clear ‘next step’ at each stage of the solution
- arrive at that solution in a finite and predictable number of steps
- ideally be well suited to computerisation.

The last point about computerisation is important because most real-world problems are too large to make a manual solution viable. However, the questions in this book and those in examinations are small in scale, so that you can solve them by hand in a reasonable time. As a result the solution may be obvious by inspection, but you must not be tempted to take short-cuts.

The examiner wants to know

- whether you know the correct algorithm to use
- whether you can follow the steps accurately to solve the problem.

You should always follow the correct steps and you must make your working clear.

Communicating an algorithm

An algorithm can be stated in two main ways.

- 1 A list of instructions.
These instructions could sometimes be in the form of a computer program.
- 2 A flowchart.

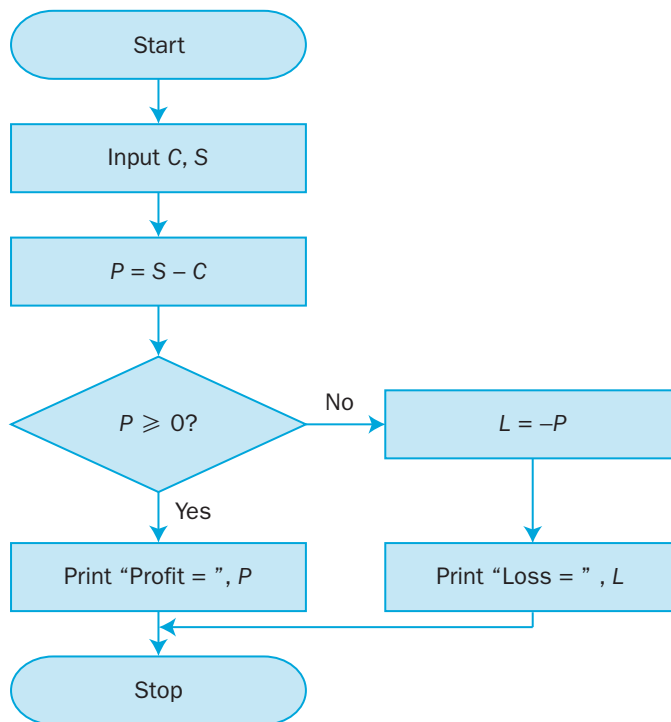
You will not be expected to write a program.

This algorithm calculates the profit or loss of a transaction, given the cost price, C , and the selling price, S .

Here is the algorithm as a list of instructions:

- Step 1** Input C, S
Step 2 Let $P = S - C$
Step 3 If $P \geq 0$ Then Print "Profit =", P : Go to Step 6
Step 4 Let $L = -P$
Step 5 Print "Loss =", L
Step 6 Stop

Here is the algorithm as a flowchart:



EXAMPLE 2

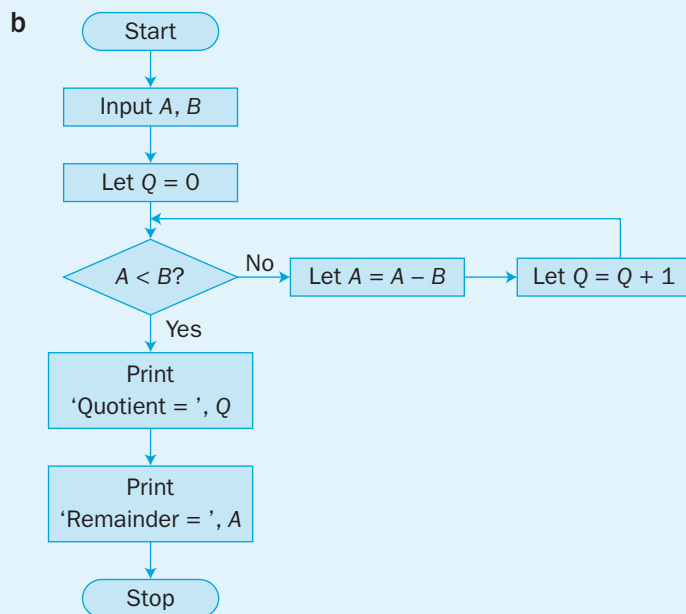
This algorithm is designed to divide a positive integer, A , by another positive integer, B , giving a quotient, Q , and a remainder.

- Step 1 Input A and B
- Step 2 Let $Q = 0$
- Step 3 If $A < B$ then go to Step 7
- Step 4 Let $A = A - B$
- Step 5 $Q = Q + 1$
- Step 6 Go to Step 3
- Step 7 Print 'Quotient = ', Q
- Step 8 Print 'Remainder = ', A
- Step 9 Stop

- a Show the operation of the algorithm when $A = 100$ and $B = 23$.
- b Display the algorithm in the form of a flowchart.

- a The best way of showing the action of the algorithm is as a table of the values it produces. This is called a **trace table**.

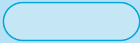


Steps	A	B	Q	Notes
1,2	100	23	0	Set up the starting values
3,4,5	77	23	1	$A > B$ so subtract B from A , increase Q by 1
6,3,4,5	54	23	2	" " " "
6,3,4,5	31	23	3	" " " "
6,3,4,5	8	23	4	" " " "
3,7,8,9	8	23	4	$A < B$ so print 'Quotient = 4, Remainder = 8'



Follow this flowchart through and satisfy yourself that it correctly describes the algorithm.

Flowchart conventions

The main points to remember when drawing a flowchart are:

- the main sequence of instructions flows down the page and from left to right
- start/stop instructions use a box like this 
- 'do this' instructions use a box like this 
- yes/no decisions use a box like this 

EXAMPLE 3

For the algorithm in the flowchart

- apply the algorithm to the list $L(1) = 8, L(2) = 11, L(3) = 6, L(4) = 14, L(5) = 10$
- describe what the algorithm achieves
- state the effect of changing the decision ' $M > L(I)$ ' to ' $M < L(I)$ '.

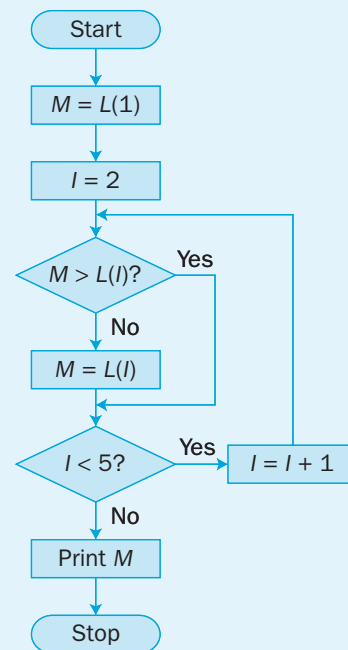
- The values taken by the variables M and I are

M	I
0	1
8	1
8	2
11	2
11	3
11	3
11	4
14	4
14	5
14	5

The variable I acts as a pointer to the position of successive values in the list.

The output value of $M = 14$.

- At each stage the algorithm remembers the larger of the two numbers. Hence the printed value of M is the largest number in the list.
- The algorithm now remembers the smaller number at each stage, so the output value of M is the smallest number in the list.



EXAMPLE 4

Susan wishes to generate a list consisting of the numbers 1 to 50 in a random order.

She proposes to use the following algorithm.

- Step 1** Use the random number function on a calculator to obtain a random number between 1 and 50
- Step 2** Write the number down unless it is already in the list
- Step 3** If the list has 50 numbers, then stop
- Step 4** Go to Step 1

Explain why this is not a viable algorithm.

An algorithm should end in a finite and predictable number of steps.

With the process described, as the list grows the chance of repeats increases. If Susan is unlucky she could go on for a very long time and there is a small probability that the number of steps could be bigger than any number you care to suggest.

Have a go at devising a better procedure for producing such a list.

D1

Exercise 1.1

- 1 List the output from this algorithm.

- Step 1** Let $A = 1, B = 1$
- Step 2** Print A, B
- Step 3** Let $C = A + B$
- Step 4** Print C
- Step 5** Let $A = B, B = C$
- Step 6** If $C < 50$ then go to Step 3
- Step 7** Stop

- 2 a List the output from this algorithm.

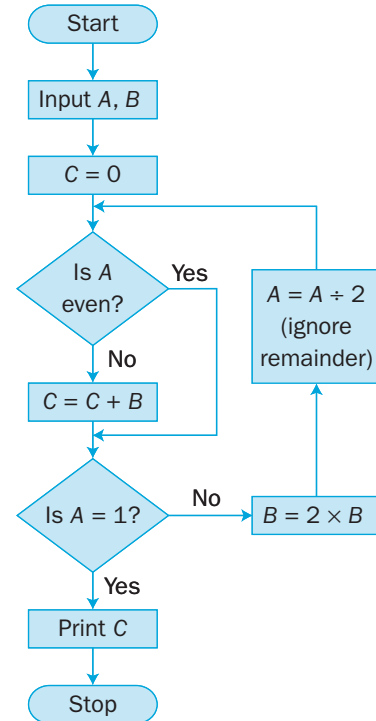
- Step 1** Let $A = 1, B = 1$
- Step 2** Print A
- Step 3** Let $A = A + 2B + 1$
- Step 4** Let $B = B + 1$
- Step 5** If $B \leq 10$ then go to Step 2
- Step 6** Stop

- b What does the algorithm achieve?

- c What would be the effect of the following errors when typing this algorithm?

- i Putting 'Print B ' instead of 'Print A '.
- ii Putting 'Let $A = A + B + 1$ ' instead of 'Let $A = A + 2B + 1$ '
- iii Putting ' $B < 10$ ' instead of ' $B \leq 10$ '

- 3 a** Apply this flowchart to
- i $A = 12, B = 7$
 - ii $A = 53, B = 76$
- b** What does the algorithm achieve?
- c** The algorithm is very inefficient when, for example, $A = 211, B = 6$.
- i Why is it inefficient?
 - ii How might you modify the flowchart to overcome this?



- 4** This is Euclid's algorithm. It finds the highest common factor of two numbers.

- Step 1 Input A, B
- Step 2 If $A \leq B$ then go to Step 4
- Step 3 Swap A and B
- Step 4 $R = \text{Remainder from } B \div A$
- Step 5 If $R = 0$ then go to Step 9
- Step 6 $B = A$
- Step 7 $A = R$
- Step 8 Go to Step 4
- Step 9 Print 'HCF = ', A
- Step 10 Stop

- a** Use the algorithm with starting values
- i $A = 48, B = 132$
 - ii $A = 130, B = 78$
- b** Draw a flowchart to show this algorithm.

- 5 This flowchart is designed to find the nature of the roots of a quadratic equation $ax^2 + bx + c = 0$ by calculating the discriminant $d = b^2 - 4ac$.

a Follow the flowchart for the equations

i $2x^2 - 6x - 3 = 0$

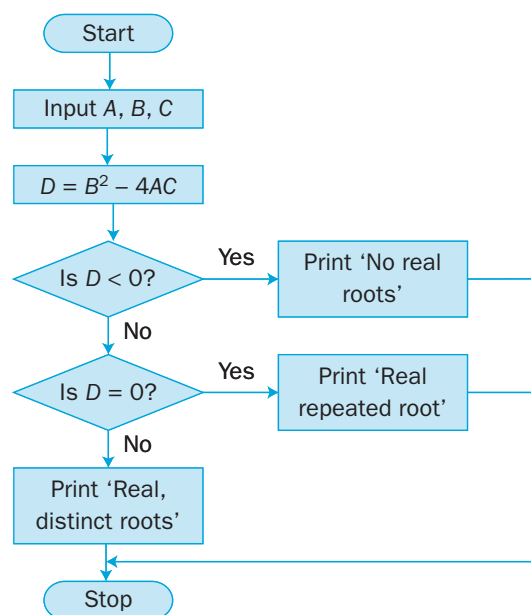
ii $3x^2 + 4x + 5 = 0$

iii $9x^2 - 6x + 1 = 0$

b The roots can be calculated, when they exist, using the quadratic formula

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Modify the flowchart so that it prints out the values of the roots where possible.



- 6 An examination consists of two papers, A and B, each marked out of 100. To gain a pass a candidate must score at least 50 on each paper, and have a total score of at least 120. To gain a distinction the total score must be at least 150. Describe, by means of a list of instructions or a flowchart, an algorithm for outputting 'Fail', 'Pass' or 'Distinction' for an input pair of marks.

- 7 A large group of adults travelling in a remote region find their route blocked by a river. They meet two children who own a boat, but the boat is only big enough to carry the two children or one adult. Nevertheless, the children manage to ferry the party across the river and return to their starting point.

- a Give a list of instructions to explain, as economically as possible, how this was done.
- b Find how many times the boat had to cross the river if there were ten adults in the party.

- 8 Draw a flowchart for an algorithm which prints out all the prime numbers between 100 and 1000.

- 9 Show that if the algorithm in Example 1 is applied to m white counters and n black counters, the puzzle will be completed in $(mn + m + n)$ moves.

1.2

Bin-packing algorithms

Some real-life problems require you to place items into compartments of a fixed size. Algorithms are used to find efficient solutions to these 'bin-packing' problems.

e.g. Loading vehicles onto a ferry with several lanes of equal length.

e.g. Suppose you wish to store items in bins 60 cm deep.

The heights, in cm, of the items are

8, 16, 12, 8, 45, 18, 30, 7, 10, 14, 9, 9, 52

The total of these heights is 238 cm, so you need at least 4 bins.

You want a packing order using the least number of bins.

Full-bin algorithm

One approach is to look for combinations of items which exactly fill a bin.

e.g. A possible solution is

Bin 1	8, 52	= 60
Bin 2	8, 7, 45	= 60
Bin 3	12, 18, 30	= 60
Bin 4	9, 9, 10, 14, 16	= 58

In this case you would need 4 bins. If the 45 and 52 had been 43 and 54, you would have needed 5 bins even though the total was unchanged at 238 cm.

DI

Full-bin algorithm:

- 1 search for combinations of values which exactly fill bins
- 2 for the remainder, place each item in the first available bin.

The problem with the full-bin algorithm is the trial-and-error nature of the first part. To do it fully you would have to examine all possible full-bin combinations to see which leads to the best solution. This is not practical for large problems and so other, more systematic procedures have been devised.

First-fit algorithm

This is essentially the second part of the full-bin algorithm.

First-fit algorithm:

taking the items in the order listed, place each item in the first bin which has room for it.

In this case the first-fit algorithm gives you a worse result, needing 5 bins. However the first-fit algorithm provides a quick and systematic way of achieving what is usually a reasonably good solution, and can be used in large problems where the full-bin approach is not suitable.

You may find it helpful to keep a record of the space left in the bins as they are filled.

8, 16, 12, 8, 45, 18, 30, 7, 10, 14, 9, 9, 52

	Contents of bin	Space left
Bin 1	8, 16, 12, 8, 7, 9	60, 52, 36, 24, 16, 9, 0
Bin 2	45, 10	60, 15, 5
Bin 3	18, 30, 9	60, 42, 12, 3
Bin 4	14	60, 46
Bin 5	52	60, 8

First-fit decreasing algorithm

This often gives a better result than the first-fit algorithm, but takes longer to process because of the need to sort the data.

First-fit decreasing algorithm:

- 1 sort the list of values into decreasing order of size
- 2 apply the first-fit algorithm.

Applying this to the example, you get

52, 45, 30, 18, 16, 14, 12, 10, 9, 9, 8, 8, 7

	Contents of bin	Space left
Bin 1	52, 8	60, 8, 0
Bin 2	45, 14	60, 15, 1
Bin 3	30, 18, 12	60, 30, 12, 0
Bin 4	16, 10, 9, 9, 8, 7	60, 44, 34, 25, 16, 8, 1

None of these algorithms guarantees to give the **optimum** (best possible) result every time. They are **heuristic** algorithms, giving an acceptably good result in a reasonable length of time. No perfect algorithm exists (yet).

EXAMPLE 1

A carpenter wants to cut 12 pieces of wood – two 40 cm long, four 60 cm long, three 80 cm long, one 120 cm long and two 140 cm long. The wood is sold in 2.4 m lengths.

- a What is the minimum number of lengths required?
- b Use the first-fit decreasing algorithm to suggest a cutting plan.
- c Show that there exists a better solution to the problem.

a The total of the required pieces is 9.6 m, so at least 4 lengths will be needed.

- b Sort the data into decreasing order and apply the first-fit procedure:

140, 140, 120, 80, 80, 80, 60, 60, 60, 60, 40, 40

	Contents of bin	Space left
Length 1	140, 80	240, 160, 20
Length 2	140, 80	240, 160, 20
Length 3	120, 80, 40	240, 120, 40, 0
Length 4	60, 60, 60, 60	240, 180, 120, 60, 0
Length 5	40	240, 200

This cutting plan requires 5 lengths and involves wastage totalling 2.4 m.

- c By inspection you can see that a more efficient solution would be

Length 1 140, 60, 40 = 240

Length 2 140, 60, 40 = 240

Length 3 80, 80, 80 = 240

Length 4 120, 60, 60 = 240

This is called the **lower bound** for the number of lengths required.

D1

You might have found this solution using the full-bin algorithm, but it would depend on the order in which you chose the full bins. e.g. Noticing immediately that the four 60 cm items 'fill a bin' would not have arrived at the best result. This shows why the full-bin approach is unsatisfactory.

Exercise 1.2

- 1** Twelve items are to be packed in bins of height 18 cm. The heights of the items, in cm, are

5, 3, 9, 6, 11, 2, 3, 7, 6, 4, 6, 10

- a** Calculate the lower bound for the number of bins involved.
- b** Use the first-fit algorithm to obtain a possible solution.
- c** Use the first-fit decreasing algorithm to obtain an improved solution.

- 2** Nineteen schools send a total of 100 students to an A-level revision day. The numbers from the schools are

5, 3, 3, 4, 6, 4, 6, 8, 10, 6, 5, 6, 2, 6, 4, 2, 8, 2, 10

The students are to be placed in workshop groups, each containing a maximum of 20 students. Students from the same school are to be in the same group.

- a** Use a full-bin approach to show that the students can fit into five workshops.
- b** Show that the first-fit algorithm does not give an optimum result.
- c** Apply the first-fit decreasing algorithm and show that it gives an optimum allocation.

- 3** A transport company is employed to move fourteen items whose masses (in tonnes) are

3.0, 1.2, 3.5, 2.5, 0.5, 1.0, 1.0, 2.5, 1.2, 0.7, 2.0, 1.8, 2.1, 1.2

The available van has a maximum payload of 5 tonnes.

- a** Use the first-fit algorithm to devise a possible loading plan. How many trips will be needed?
- b** Use the first-fit decreasing algorithm to obtain an improved plan. How many trips are needed with this plan?
- c** Explain why it is not possible to do the job in fewer trips than this.

- 4 An upholsterer needs to cut various lengths of material from standard 12 m rolls. The lengths required, in metres, are

2, 2, 3, 3, 4, 6, 7, 9

- a Show that neither the first-fit nor the first-fit decreasing algorithm gives a solution using less than four rolls.
- b Show that it is possible to solve the problem using only three rolls.

- 5 A music fan stores CDs in boxes with a capacity of 15 discs. She classifies the CDs according to the artiste, and intends to ensure that all CDs by a given artiste are stored in the same box. The numbers of CDs in each classification are

6, 11, 4, 12, 9, 3, 5, 5, 6, 6, 2, 8, 9

Using a suitable algorithm and showing all your working, investigate whether she can store her CDs in six boxes.

- 6 A company has a number of tasks to be completed in a 35-hour week. Each task is to be done from start to finish by one person. The projected lengths of the tasks are shown in the table. The company plans to hire temporary staff for these tasks.

No of hours	4	5	6	7	8	9
No of tasks	8	10	4	6	3	4

- a Show that a task allocation produced using the first-fit decreasing algorithm would require the company to hire seven staff.
- b Show that by modifying the allocation in part a the company need only hire six staff.

- 7 The organiser of a local talent contest wishes to arrange the evening as three 45-minute sections with drink and chat intervals between. There are ten acts in the competition with running times as shown (in minutes).

6, 8, 10, 20, 6, 10, 24, 15, 20, 15

- a Show that the first-fit algorithm fails to give a suitable running order.
- b Use the first-fit decreasing algorithm to find a running order.
- c The organiser decides that there should be at least three acts in each section. Modify your answer to part b to allow for this.

1.3 Sorting algorithms

One of the most common data processing tasks is sorting an unordered list into numerical or alphabetical order. For long lists and for computer implementation this requires an algorithm.

Sorting a long list can take a lot of processing time. Many sorting algorithms have been devised. You need to know the details of two – the bubble sort and the quick sort.

Bubble sort

Bubble sort

- **First pass**

Compare the 1st and 2nd numbers. Swap them if in the wrong order.

Compare/swap the 2nd and 3rd numbers, then the 3rd and 4th numbers, and so on to the end of the list.

- **Subsequent passes**

Repeat the pass process, leaving out the final number from the previous pass.

- **Terminating**

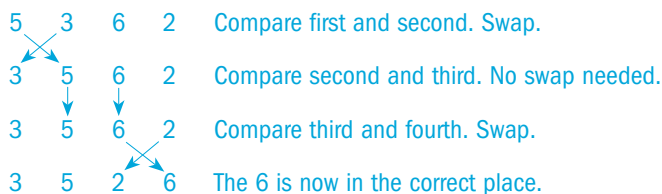
Finish when either the last pass had only two numbers or a complete pass produced no swaps.

Short lists, such as those in exam questions, can be easily sorted 'by inspection'. Remember that the question is testing whether you know the algorithm, not whether you can sort the list.

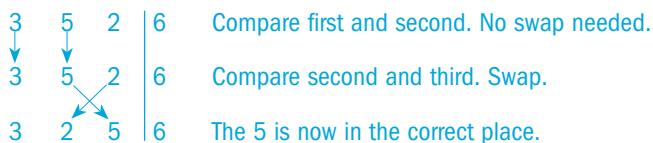
If you search the internet you will find twenty or more sorting algorithms. They vary enormously in speed, and there is no single best method.

DI

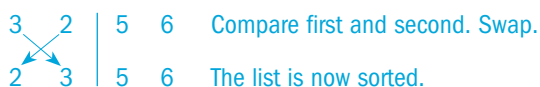
e.g. Sorting the list 5, 3, 6, 2 into ascending order, the first pass looks like this:



Because the final number is now in the right place, you can ignore it on the second pass.



On the next pass you can ignore the final two numbers.



If at any stage you make a complete pass without doing any swaps, it means the list is in the correct order, so you stop.

EXAMPLE 1

Arrange the numbers 4, 8, 2, 6, 3, 5 in ascending order using the bubble sort algorithm. Record the number of comparisons and the number of swaps made at each stage.

First pass

4	8	2	6	3	5
4	8	↔	2	6	3
4	2	8	↔	6	3
4	2	6	8	↔	3
4	2	6	3	8	↔
4	2	6	3	5	8

The first pass involved 5 comparisons and 4 swaps.

Second pass

4	↔	2	6	3	5	8
2	4	6	3	5	8	
2	4	6	↔	3	5	8
2	4	3	6	↔	5	8
2	4	3	5	6	8	

The second pass involved 4 comparisons and 3 swaps.

Third pass

2	4	3	5	6	8
2	4	↔	3	5	6
2	3	4	5	6	8
2	3	4	5	6	8

The third pass involved 3 comparisons and 1 swap.

Fourth pass

2	3	4	5	6	8
2	3	4	5	6	8
2	3	4	5	6	8

The fourth pass involved 2 comparisons and 0 swaps.

In the tables, the shading shows the comparison being made at each stage, with the arrow indicating if a swap is necessary.

The 8 has moved to its correct place after this pass.

The 6 has moved to its correct place after this pass.

The 5 has moved to its correct place after this pass.

There were no swaps on this pass, so the sort is complete.

Different sorting algorithms vary in how efficiently they accomplish the task. This is measured by how many operations are involved, which in the case of the bubble sort is the number of comparisons and swaps made.

In Example 1 the bubble sort involved a total of 14 comparisons and 8 swaps.

If a computer is being used, this translates into the amount of processing time needed.

Quick sort

This algorithm takes a different approach from the comparison/swap method of the bubble sort.

One number is chosen to be the **pivot**. The rest are split into two sub-lists – numbers less than the pivot and numbers greater than the pivot. The procedure is then repeated for each sub-list, with its own pivot, until the list is sorted.

The list starts in random order, so the choice of pivot is arbitrary. Some authors use the first value in the list, but this syllabus will expect you to use the middle value in the list. If there are two ‘middle’ values, either can be used – this book will use the second one as the pivot.

Quick sort

- Step 1** Choose the middle number as the pivot.
Step 2 For each remaining number in the list:
 if number \leq pivot, place number in sub-list before pivot; otherwise place number in sub-list after pivot.
Step 3 **Quick sort** sub-lists containing two or more numbers

The quick sort can be defined in compare/swap terms, but this lies outside the present syllabus.

If a number is equal to the pivot, you could put it in either sub-list. In this book the ‘equal’ numbers are included in the ‘less than’ sub-list.

This is in line with the practice used in binary search – see Section 1.4.

Numbers go in the sub-lists in the order in which they appeared in the original list. (This is an arbitrary choice, but an algorithm must be definite at each stage.)

The quick sort is an example of a **recursive algorithm**, that is, one which contains itself.

The 5 is the second of the two middle numbers.

The 1 is the second of the two middle numbers.

To bubble sort this list you would have made 45 comparisons, whereas the quick sort involved just 20.

Arrange the numbers 4, 8, 3, 1, 7, 5, 12, 13, 2, 6 in ascending order using the quick sort algorithm.

4	8	3	1	7	5	12	13	2	6
---	---	---	---	---	---	----	----	---	---

Taking the 5 as the pivot gives these sub-lists.

4	3	1	2	5	8	7	12	13	6
---	---	---	---	---	---	---	----	----	---

The pivots for the two sub-lists are 1 and 12, giving

1	4	3	2	5	8	7	6	12	13
---	---	---	---	---	---	---	---	----	----

Two sub-lists have two or more numbers. The pivots are 3 and 7.

1	2	3	4	5	6	7	8	12	13
---	---	---	---	---	---	---	---	----	----

The remaining sub-lists each have length 1, so the process is complete.

The quick sort is usually more efficient than the bubble sort.

Exercise 1.3

1 Use the bubble sort algorithm to sort this list into

- a** ascending order **b** descending order.

12 4 16 5 9 2 4

Show the result of each pass.

- 2 Repeat question 1 using the quick sort algorithm.
- 3 Sort this list into ascending order using the bubble sort algorithm.

22 26 14 20 12 9 11 15 10

- 4 Sort this list into ascending order using the quick sort algorithm.

9 17 6 19 16 13 7 17 12 9

- 5 A class has students with surnames

Harris, Thomas, Patel, Frobisher, Cheung, Allen and Lee.

Sort these into alphabetical order using

- a bubble sort b quick sort.

- 6 Another sorting algorithm is the interchange sort. For ascending order the algorithm is:

Step 1 Find the largest number. Swap it (if necessary) with the last number

Step 2 Ignoring the last number, repeat Step 1 with the remaining list

Step 3 Repeat Step 2 until there is only one number in the remaining list

- a Use the interchange sort to arrange the list 1, 5, 9, 3, 11, 7, 13 in ascending order. Record the state of the list after each step, and count the number of comparisons and swaps needed.
- b Sort the same list using the bubble sort. Which method was more efficient?
- 7 The worst-case scenario for a bubble sort is that the original list is in the reverse of the required order.
- a For this worst case, find the number of comparisons and swaps when sorting a list of five numbers.
- b How many would be needed for a list of
i 10 ii 20 iii n numbers?

- 8 The worst-case scenario for a quick sort depends on the convention being used for choosing the pivot. For a quick sort using the middle number convention

a give a worst case starting order for the numbers
1, 2, 3, 4, 5, 6, 7

b find the number of comparisons (between numbers and the pivot) which would be needed in this case.

When you ask a computer to search a database for an item (which may or may not be there), the computer does not check every entry in the file. It uses an algorithm to reduce the number of items it needs to examine.

Here is an illustration of the most common search algorithm:

Your friend chooses a number between 1 and 20, and asks you to identify it by asking as few questions as possible.

You could just keep guessing different numbers until you hit on the right one. However, unless you strike lucky, the size of the problem only reduces by one at each attempt. If you were unlucky, you could take 19 questions.

The best method is to ask how the number compares (equal, less or greater) with the middle number of the list. Either the number is found or the problem size is halved.

Suppose your friend chooses 14. The middle number is 11.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----

Your friend tells you ‘The number is above 11’.

The number lies between 12 and 20. The middle number is 16.

12	13	14	15	16	17	18	19	20
----	----	----	----	----	----	----	----	----

Your friend tells you ‘The number is below 16’.

The number lies between 12 and 15. The middle number is 14.

12	13	14	15
----	----	----	----

Your friend tells you ‘The number is equal to 14’.

You have found the number in three attempts.

As in Section 1.3, if there are two ‘middle’ values choose the second of these.

11 is the second of the two middle numbers.

14 is the second of the two middle numbers.

The most attempts you would have needed is four (e.g. if your friend had chosen 13).

This process is called a **binary search**. It is widely used for searching sorted lists of data.

The list to be searched must be sorted into order (numerical or alphabetical) before a binary search can be done.

Binary search

To search a list for the item '*TARGET*'

- Step 1** Find middle item, M , of search list.
- Step 2** If $TARGET = M$ then $TARGET$ has been found: Stop.
- Step 3** If $TARGET < M$ then new search list is items below M .
If $TARGET > M$ then new search list is items above M .
- Step 4** Repeat from Step 1 until $TARGET$ is found or search list is empty ($TARGET$ not in list).

The difference between a general binary search and the number guessing game is that the target item may not appear in the list.

You need to be able to find the middle position of a search list.

For a search list starting at position n_1 and ending at position n_2 :

if the number of items is odd, the middle position is

$$m = \frac{1}{2}(n_1 + n_2)$$

if the number of items is even, the middle position is the next integer value above this (the second of the two middle items).

Notation: $\left\lceil \frac{1}{2}(n_1 + n_2) \right\rceil$ means 'the smallest integer greater than or equal to $\frac{1}{2}(n_1 + n_2)$ '.

The middle position of a search list from position 12 to position 20 is $\left\lceil \frac{1}{2}(12 + 20) \right\rceil = 16$.

The middle position of a search list from position 15 to position 24 is $\left\lceil \frac{1}{2}(15 + 24) \right\rceil = \left\lceil 19\frac{1}{2} \right\rceil = 20$.

Perform a binary search on the following list to look for

a Earle **b** Underwood.

- | | | | |
|---------------|--------------|--------------|------------|
| 1. Asquith | 2. Auden | 3. Berisford | 4. Dubarry |
| 5. Earle | 6. Ford | 7. Harris | 8. Khan |
| 9. Martin | 10. O'Malley | 11. Treacher | 12. Wilson |
| 13. Zuckerman | | | |

a The middle position in the list is $\left\lfloor \frac{1}{2}(1 + 13) \right\rfloor = 7$,
which is Harris.

Earle comes before Harris, so the new search list is

- | | | |
|------------|----------|--------------|
| 1. Asquith | 2. Auden | 3. Berisford |
| 4. Dubarry | 5. Earle | 6. Ford |

The middle position of this list is $\left\lfloor \frac{1}{2}(1 + 6) \right\rfloor = \left\lfloor 3\frac{1}{2} \right\rfloor = 3$,
which is Dubarry.

Earle comes after Dubarry, so the new search list is

- | | |
|----------|---------|
| 5. Earle | 6. Ford |
|----------|---------|

The middle position of this list is $\left\lfloor \frac{1}{2}(5 + 6) \right\rfloor = \left\lfloor 5\frac{1}{2} \right\rfloor = 5$,
which is Ford.

Earle comes before Ford, so the new search list is just

- | |
|----------|
| 5. Earle |
|----------|

The middle of this list is obviously position 5, which is Earle. The algorithm has located Earle in position 5.

b The middle position in the list is $\left\lfloor \frac{1}{2}(1 + 13) \right\rfloor = 7$,
which is Harris.

Underwood comes after Harris, so the new search list is

- | | | |
|--------------|------------|---------------|
| 8. Khan | 9. Martin | 10. O'Malley |
| 11. Treacher | 12. Wilson | 13. Zuckerman |

The middle position of this list is $\left\lfloor \frac{1}{2}(8 + 13) \right\rfloor = \left\lfloor 10\frac{1}{2} \right\rfloor = 10$,
which is Treacher.

Underwood comes after Treacher, so the new search list is

- | | |
|------------|---------------|
| 12. Wilson | 13. Zuckerman |
|------------|---------------|

The middle position of this list is $\left\lfloor \frac{1}{2}(12 + 13) \right\rfloor = \left\lfloor 12\frac{1}{2} \right\rfloor = 12$,
which is Zuckerman.

Underwood comes before Zuckerman, so the new search list is just 12. Wilson

The middle of this list is obviously position 12, which is Wilson.

Underwood comes before Wilson, but the list is now empty, so Underwood is not on the list.

Remember that it is your understanding of the binary search process that is being tested. It is important to show details of your working.

Exercise 1.4

For binary search questions, your working should show the sequence of items examined during the search.

1 Identify the middle item in these lists.

- | | |
|---|--|
| <p>a 1 – Aktar
2 – Amner
3 – Charles
4 – Ferriday
5 – Garton
6 – Nish
7 – Patel
8 – Quedgeley</p> | <p>b 9 – Dewsbury
10 – Evesham
11 – Exeter
12 – Garstang
13 – Holmfirth
14 – Jarrow</p> |
| <p>c 16 – Fiat
17 – Ford
18 – Hyundai
19 – Jaguar
20 – Kia
21 – Lada
22 – Lotus
23 – Nissan
24 – Seat
25 – Skoda
26 – Toyota</p> | |

2 Use binary search to look for

- a** Nish in the list from question 1a
b Exhall in the list from question 1b
c Jaguar in the list from question 1c.

3 The table shows the names of children booked for a coach trip.

No	Name	No	Name	No	Name	No	Name
1	Barry	6	Edith	11	Laurence	16	Quentin
2	Boris	7	Floella	12	Mabel	17	Rebecca
3	Catherine	8	Gerald	13	Nuria	18	Tristan
4	Cedric	9	Ingrid	14	Omar	19	Xavier
5	Declan	10	Juan	15	Petula	20	Zoe

Use binary search to look for

- a** Omar **b** Floella **c** Mary.

- 4 A company's database is set up so that a customer's record can be found by entering their postcode. The table shows the list. Use binary search to find the name of the customer whose postcode is

a BS7 8NB b TA6 8KC.

No	Postcode	Name	No	Postcode	Name	No	Postcode	Name
1	BA4 6AS	L. Muswell	11	BS6 2JG	S. Voce	21	TA1 5HA	E. O'Flynn
2	BA4 7TS	K. Bennett	12	BS7 3LE	O. Banwell	22	TA3 8FF	U. Paine
3	BA4 7VA	A. Thorpe	13	BS7 8NB	I. R. Smith	23	TA3 9BS	T. Dearing
4	BA5 3HG	B. Fitzroy	14	BS9 6BS	F. Button	24	TA4 4AT	B. I. Twist
5	BA6 9FR	G. Mander	15	BS12 2IJ	D. Aire	25	TA4 7NS	V. Tingle
6	BA11 2JJ	R. Singh	16	BS17 3YY	W. Jones	26	TA5 2MN	D. Harold
7	BA12 1PJ	I. West	17	BS21 9AB	J. Kryzwicki	27	TA6 8KC	S. Elliott
8	BA22 7JP	P. D. Quicke	18	BS21 9CE	P. Gomez	28	TA6 8KT	C. Pericles
9	BS3 4HR	N. Ambrose	19	TA1 4EF	H. Smith-Ball	29	TA8 3DD	Y. Nott
10	BS3 4HW	B. Fouracres	20	TA1 5GK	A. McKinley	30	TA8 4YD	R. Taylor

- 5 A market research company plans to send interviewers to a random sample of 40 houses in a city street of 300 houses. The owner of number 227 contacts them to ask if her house is involved. Perform a binary search of this list of house numbers to answer her query.

Interview no.	1	2	3	4	5	6	7	8	9	10
House no.	14	55	64	69	70	71	83	120	121	129
Interview no.	11	12	13	14	15	16	17	18	19	20
House no.	141	143	145	152	154	163	166	183	186	188
Interview no.	21	22	23	24	25	26	27	28	29	30
House no.	189	192	198	200	202	203	204	209	220	229
Interview no.	31	32	33	34	35	36	37	38	39	40
House no.	231	232	234	241	252	274	278	280	285	287

Review 1

1 a List the output from this algorithm.

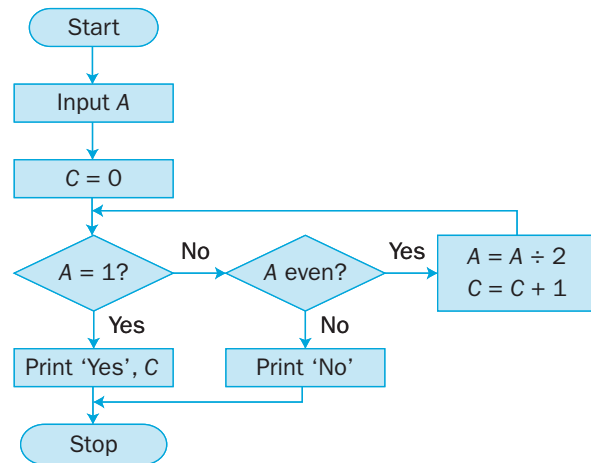
- Step 1 $A = 0, B = 1, C = 0$
 Step 2 $A = A + B$
 Step 3 Print A
 Step 4 $C = C + 6$
 Step 5 $B = B + C$
 Step 6 If $A < 500$ then go to Step 2
 Step 7 Stop

b What does the algorithm achieve?

2 a Execute the algorithm shown in the flowchart for each of these input values.

- i 12 ii 8 iii 30 iv 32

In each case list the values taken by A and C as the algorithm proceeds, together with the output.



b Explain what the algorithm achieves.

3 a What would be the output from this algorithm?

- Step 1 $A = 1, B = 1$
 Step 2 Print A, B
 Step 3 $C = A + B$
 Step 4 Print C
 Step 5 $A = B, B = C$
 Step 6 If $C < 50$ go to Step 3
 Step 7 Stop

b Draw a flowchart to show the algorithm from part a.

4 Use the quick sort algorithm to sort these children into increasing order of age.

Name	Chloe	Adam	Dan	Sally	Miguel	Karl	Tanya	Usha	Pete	Fay
Age (years)	8	5	15	7	13	9	12	6	9	10

- 5 A ferry company has a small vehicle ferry with four lanes, each 40 m long. The spaces in metres needed for the vehicles on a certain day (in order of arrival) were

4, 6, 5, 12, 14, 5, 5, 6, 4, 4, 14, 16, 10, 5, 6, 5, 8, 13, 6, 4, 4

- a Use the first-fit algorithm to place them in the lanes and record what you find.
- b It is then suggested that the approach is amended to

As vehicles arrive, line them up in two queues, one for vehicles 10 m or over, the other for shorter vehicles. Load first the long vehicles, then the short vehicles, using the first-fit algorithm.

What would be the result of using this modified approach for the vehicles listed above?

- 6 A cable car has space for 15 passengers. Passengers arrive in travel groups. At the start of the day there is a long queue, with the number of people in each travel group as shown.

9 4 3 8 6 5 9 6 5 4 8 4 3

The members of each travel group wish to stay together.

- a Find the lower bound for the number of trips that might be required to clear the backlog.
- b Show that the first-fit decreasing algorithm does not give a solution involving this minimum number of trips.
- c Find by inspection a plan which does the job in the minimum number of trips.
- 7 A plumber buys copper pipe in 4 m lengths. For a particular job he requires the lengths shown.

Length of pipe (m)	0.3	0.4	0.5	0.6	0.7	1.2	1.5	2.0
Number required	2	3	1	2	3	2	1	3

Devise a cutting plan to do the job with as little waste as possible.

- 8 Use the bubble sort algorithm to sort this list into ascending order. Show the state of the list after each pass.

12 4 9 15 11 12 8 2 7

9 Repeat question 7 using the quick sort algorithm, again showing the list after each stage.

10 Use the bubble sort algorithm to sort this list of surnames into alphabetical order.

Hughes, Benn, Carter, Lee, Jenner, Afsar, Brown, Burns, Wilson, Bligh

11 Another method of sorting is using the shuttle sort algorithm:

First pass

Compare the 1st and 2nd numbers. Swap them if they are in the wrong order.

Second pass

Compare the 2nd and 3rd numbers. Swap them if they are in the wrong order.

If they were swapped, redo the first pass.

Third pass

Compare the 3rd and 4th numbers. Swap them if they are in the wrong order. If they were swapped, redo the second pass.

Subsequent passes

Repeat the above process, introducing the next number in the list at each stage. You backtrack through the list, comparing and swapping until no swap occurs.

Terminating

A list of n items terminates after $(n - 1)$ passes.

a Use the shuttle sort algorithm to sort this list into ascending order. Find the total number of comparisons and swaps made.

125 86 315 97 130 266 45 120 284 90

b Sort the same list using the bubble sort. Find the total number of comparisons and swaps made.

12 This list of numbers is to be sorted into descending order.

55 80 25 84 25 34 17 75 3 5

a Perform a bubble sort to obtain the sorted list, giving the state of the list after each complete pass.

The numbers in the list represent weights, in grams, of objects which are to be packed into bins that hold up to 100 g.

b Determine the least number of bins needed.

c Use the first-fit decreasing algorithm to fit the objects into bins which hold up to 100 g.

[(c) Edexcel Limited 2002]

- 13** Perform a binary search to look for each of the following names in the list:

a Simpson	b Harman	c Timms	
1 – Berisford	5 – Harris	9 – Mountjoy	13 – Trueman
2 – Cheung	6 – Ingoe	10 – Ng	14 – Unwin
3 – Faruq	7 – James	11 – Simpson	15 – Wyndham
4 – Harman	8 – MacAllister	12 – Thompson	16 – Young

- 14** An ornithologist makes a list of all the species seen during a bird watching trip.

No	Species	No	Species	No	Species	No	Species	No	Species
1	Blue tit	6	Egret	11	Jay	16	Pied wagtail	21	Swallow
2	Brambling	7	Goldfinch	12	Kingfisher	17	Robin	22	Whitethroat
3	Bullfinch	8	Great tit	13	Lapwing	18	Song thrush	23	Widgeon
4	Buzzard	9	Greenfinch	14	Nuthatch	19	Sparrowhawk	24	Wren
5	Dunnock	10	Heron	15	Pheasant	20	Starling	25	Yellowhammer

- a** Use the binary search algorithm to look for these birds in the list:

i Heron ii Nuthatch iii Swift

In each case state the number of records which had to be examined.

- b** A linear search involves examining every record in turn until the required record is found. How many records would have been examined if this technique had been used instead of binary search in part a?

- 15** 1. Glasgow 6. Birmingham
 2. Newcastle 7. Cardiff
 3. Manchester 8. Exeter
 4. York 9. Southampton
 5. Leicester 10. Plymouth

A binary search is to be performed on this list to locate the name Newcastle.

- a** Explain why a binary search cannot be performed with the list in its present form.
- b** Using an appropriate algorithm, alter the list so that a binary search can be performed. State the name of the algorithm you use.
- c** Use the binary search algorithm on your new list to locate the name Newcastle.



Summary

Refer to

- An **algorithm** is a set of instructions, described by a list or a flowchart, for solving all problems of a given type. 1.1
 It should
 - provide a clear ‘next step’ at each stage of the solution
 - arrive at that solution in a finite and predictable number of steps
 - ideally be well suited to computerisation.
- Bin-packing algorithms – **full-bin**, **first-fit**, **first-fit decreasing** – are used to allocate items to groups of a fixed size. 1.2
- Sorting algorithms – **bubble sort**, **quick sort** – are used to sort lists of items into numerical or alphabetical order. 1.3
- The **binary search** algorithm is used to search an ordered list for a particular item. 1.4

Links

This chapter has included several examples of how sorting algorithms can be used to solve one-dimensional problems in real life.

The bin-packing algorithm has many applications such as scheduling lectures of different lengths into a given number of lecture theatres or compressing data files on a hard disk into clusters of a fixed size.

There is still much research ongoing into algorithms for two- and three-dimensional bin packing problems. Most algorithms rely to some extent on the first-fit principle.

An example of a two-dimensional problem is cutting rectangles of different sizes from standard sheets of metal, with the aim of minimising wastage.

Three-dimensional problems include loading pallets onto lorries or packages into shipping containers. Such problems may have additional constraints to do with the weight and balance of the load.

