

1A Understanding Algorithms

Magic Trick

1. **Pick** any number.
2. **Double** it.
3. **Add** 10 to the total.
4. **Divide** the result by 2.
5. **Subtract** the original number.

5?

1. Happy numbers:
 - Write down any integer
 - Square its digits and find the sum of the squares
 - Repeat this process with the new number until you get 1 (happy) or identify a cycle (unhappy)
- a) Show that 70 is happy

-
- b) Show that 4 is unhappy

2.

a) Implement this algorithm using a trace table

1. Let $n=1$, $A=1$, $B=1$
2. Print A and B
3. Let $C=A+B$
4. Print C
5. Let $n=n+1$, $A=B$, $B=C$
6. If $n<5$, go to 3.
7. If $n=5$, stop.

Instruction Step	n	A	B	C	Print

b) Describe the numbers generated by the algorithm

3. The algorithm multiplies the two numbers A and B

- 1** Make a table with two columns.
Write A in the top row of the left-hand column and B in the top row of the right-hand column.
- 2** In the next row of the table write:
 - in the left-hand column, the number that is half of A , ignoring remainders
 - in the right-hand column, the number that is double B
- 3** Repeat step 2 until you reach the row which has a 1 in the left-hand column.
- 4** Delete all rows where the number in the left-hand column is even.
- 5** Find the sum of the non-deleted numbers in the right-hand column.
This is the product AB .

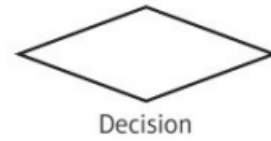
This famous algorithm is sometimes called 'the Russian peasant's algorithm' or 'the Egyptian multiplication algorithm'.

Implement this algorithm for:

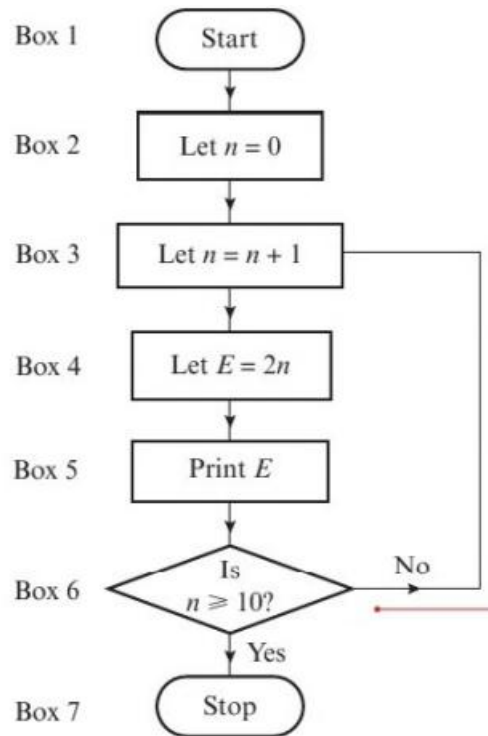
a) $A=29$ and $B=34$

b) $A=66$ and $B=56$

1B Flow Charts



1.



a) Implement this algorithm using a trace table

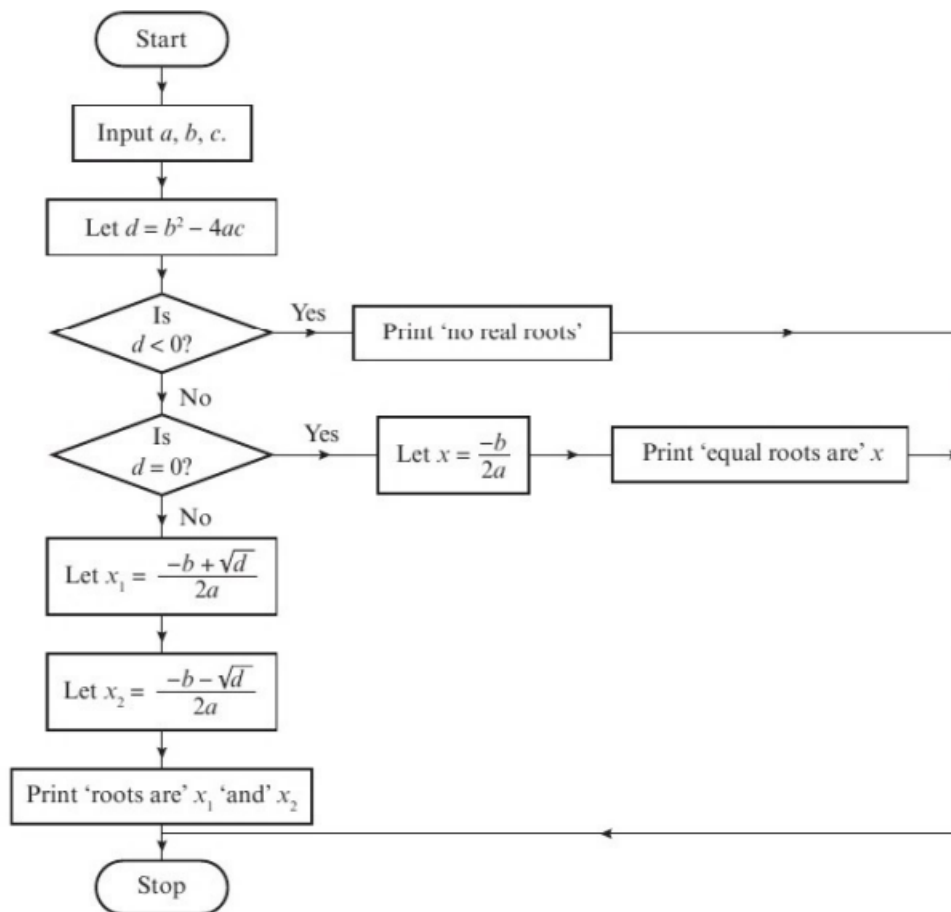
[illegible]

b) Alter box 4 to read “let $E=3n$ ” and implement the algorithm again. What do you notice?

[illegible]

2.

This flow chart can be used to find the roots of an equation of the form $ax^2 + bx + c = 0$



Demonstrate this algorithm for these equations:

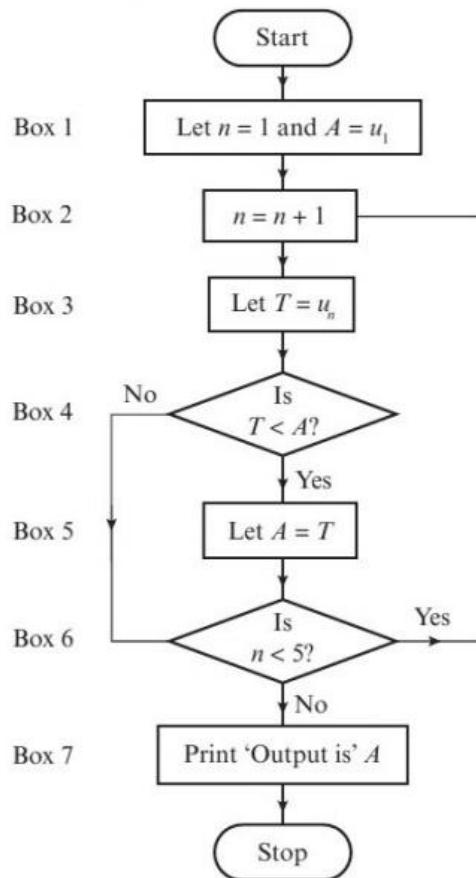
a) $6x^2 - 5x - 11 = 0$

b) $x^2 - 6x + 9 = 0$

c) $4x^2 + 3x + 8 = 0$

3. Apply the algorithm shown by the flow chart to the data

$$u_1 = 10, u_2 = 15, u_3 = 9, u_4 = 7, u_5 = 11,$$

[illegible]

1C Bubble Sort

For organising data numerically or alphabetically.

- Compare two adjacent items in a list
- If they are in order leave them and compare the next pair
- If they are not in order, swap them
- Finish the pass and repeat
- The list is in order when a pass is completed without any swaps

1. Use a bubble sort to arrange this list into ascending order

24 18 37 11 15 30

2. A list of n numbers is to be sorted into alphabetical order, starting at the left hand end of the list.
- a) Describe how to carry out the first pass of a bubble sort on the letters in the list

- b) Carry out the first pass of a bubble sort to arrange the letters in the word ALGORITHM into alphabetical order, showing every step of working.

- c) Complete a second pass to show the order of letters at the end of a second pass.

3. Use a bubble sort to arrange these numbers into descending order.

63 48 57 55 32 48 72 49 61 39

1D Quick Sort

- Select “pivot” points at the mid-point of the data list (pick to right if even)
- Move all items less to one side and all items above to the other, retaining order
- Repeat for each list either side of the original pivot (the number of pivots can exponentially grow)
- Stop when every item has been a pivot.

1. Use quick sort to arrange the numbers below into ascending order.

21 24 42 29 23 13 8 39 38

2. Use quick sort to arrange the numbers below into descending order.

37 20 17 26 44 41 27 28 50 17

1E Bin-packing

The process of splitting differently sized items into equally sized “bins”

Finding the Lower Bound

1. Nine boxes of a fixed cross-section have heights, in meters, as follows.

0.3 0.7 0.8 0.8 1.0 1.1 1.1 1.2 1.5

They are to be packed into bins with the same fixed cross-section and height 2m.

Determine the lower bound of bins needed.

First-Fit

- Take the items in the order given (even if not already in ascending)
- Place each item in the first available bin that can take it (starting from bin 1 each time)

First-Fit Decreasing

- Sort items into descending order
- Apply the first-fit algorithm

Full-Bin Packing

- Use observation to find combinations of items that fill a bin
- Use first-fit with any remaining items

2. Use the first-fit algorithm to pack the following items into bins of size 20 (the numbers in brackets are the size of the item). State the number of bins used and the amount of wasted space

A(8) B(7) C(14) D(9) E(6) F(9) G(5) H(15) I(6) J(7) K(8)

3. Apply the first-fit decreasing algorithm to the data in example 2.

4. The items below are packed into bins of size 25.

A(8) B(7) C(10) D(11) E(13) F(17) G(4) H(6) I(12) J(14) K(9)

a) Determine the lower bound for the number of bins

b) Apply the full-bin algorithm

c) Is your solution optimal? Give a reason for your answer.

5. A plumber needs to cut the following lengths of copper wire (lengths in metres).

A(0.8) B(0.8) C(1.4) D(1.1) E(1.3) F(0.9) G(0.8) H(0.9) I(0.8) J(0.9)

The pipe comes in lengths of 2.5m.

- a) Calculate the lower bound for the number of lengths of pipe needed.
- b) Use the first fit decreasing algorithm to determine how the required lengths may be cut from the 2.5m lengths.
- c) Use full bin packing to find an optimal solution.

1F Order of an Algorithm/ Run Time

How changes in size affect time taken.

Bubble sort

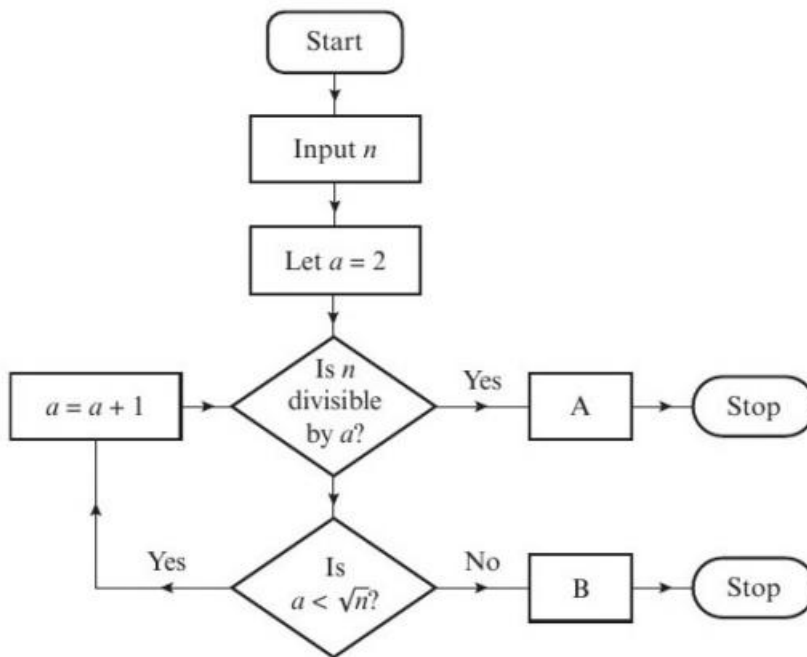
If a bubble sort has n items, how many comparisons are made on the first pass?

Given at least one item has been sorted, how many potential comparisons could be made on the second pass?

What would the total number of comparisons potentially be?

The order of an algorithm can be described as a function of its size. If an algorithm has order $f(n)$ then increasing the size of the problem from n to m will increase the run time by a factor of approximately $\frac{f(m)}{f(n)}$

1. An algorithm is defined by the below flow diagram, where $n > 2$ and n is an integer.



- a) Describe what the algorithm does
- b) Suggest suitable output text for boxes A and B.
- c) Determine the order of the algorithm (how many potential steps are there?).

In general, if the size of a problem is increased by some factor k :

- An algorithm of linear order will take approximately k times as long to complete
- An algorithm of quadratic order will take approximately k^2 times as long to complete
- An algorithm of cubic order will take approximately k^3 times as long to complete

2. A given list-searching algorithm has linear order.

To search a list with 500 values, the algorithm takes approximately 0.036 seconds.

Estimate the time taken by this algorithm to search a list containing 1250 values.

3. The bubble sort algorithm has quadratic order

- a) Given that it takes a computer 0.2 seconds to sort a list of 1600 numbers using the bubble sort, estimate the time needed to sort a list of 480000 numbers.

- b) Comment on the suitability of the bubble sort algorithm for sorting large data sets.