# Algorithms - Exercises (with Sol'ns) (24 pages; 18/2/26)

## Purposes of algorithms

(1)(i) By performing traces, or otherwise, establish what the following algorithm achieves.

10 Input $N$

20 $e = 0.0001$

30 $L = 1000$

40 $F = 0$

50 $x = \frac{N}{2}$

60 $y = \frac{N}{x}$

70 $z = x$

80 $x = \frac{x+y}{2}$

90 If $|x - z| < e$ Then Goto 130

100 $F = F + 1$

110 If $F > L$ Then Goto 140

120 Goto 60

130 Print $x$

140 Print "End"

150 END

(ii) What roles do $e$, F and $L$ play in the algorithm?

## Solution

(i) The algorithm finds the square root of $N$.

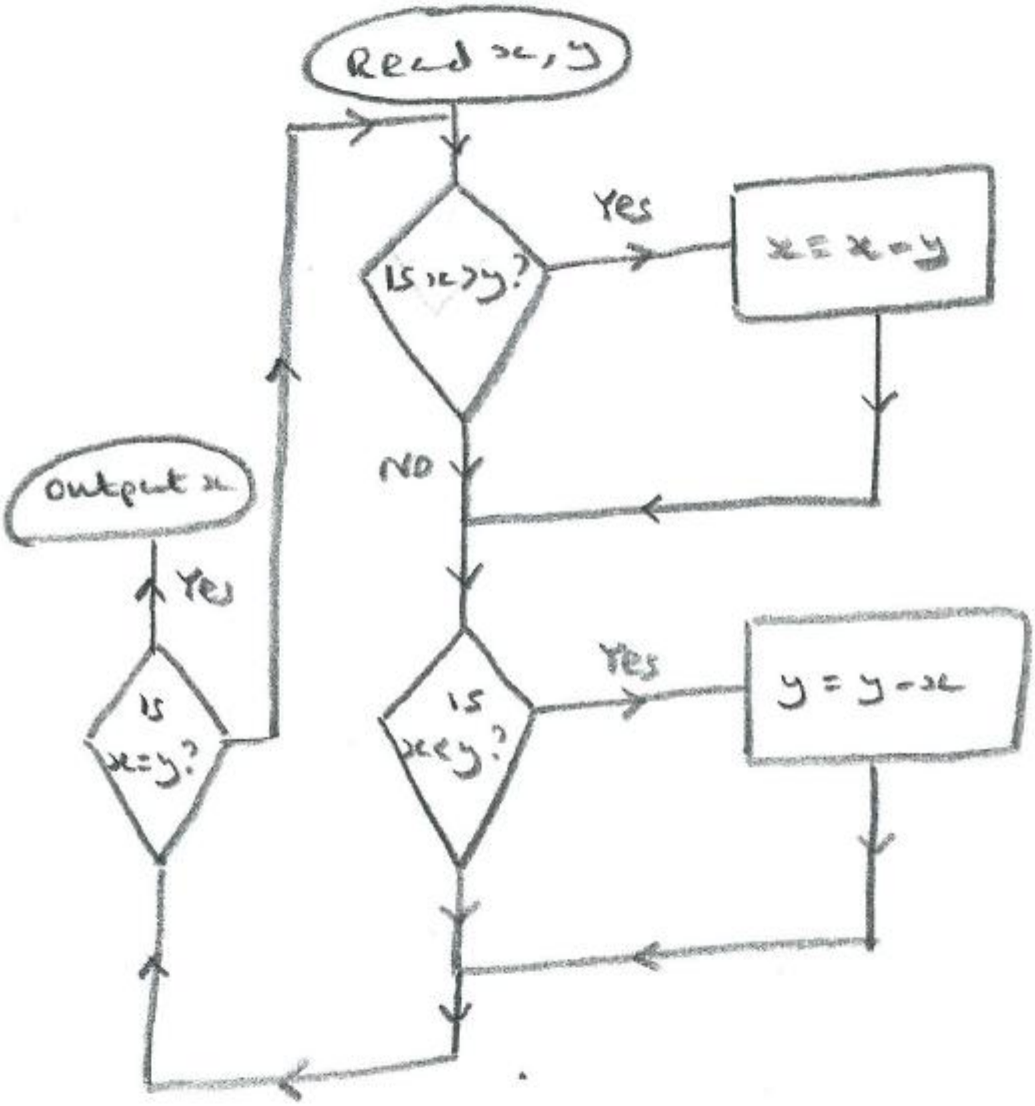
(ii) $e$ determines when the successive estimates are close enough together

$F$ counts the number of iterations

$L$ is the limit imposed on the number of iterations

[Note that $e$ and $L$ are 'hard-coded', to avoid having too many inputs, but that their values are stored at the start of the program, so that changes can easily be made.]

(2) By performing traces, or otherwise, establish what the following algorithm achieves.



**Solution**

For example, initial values: $x = 20, y = 45$

Is $x > y$? No

Is $x < y$? Yes

$y = y - x = 45 - 20 = 25$ $[x = 20, y = 25]$

Is $x = y$? No

Is $x > y$? No

Is $x < y$? Yes

$y = y - x = 25 - 20 = 5$ $[x = 20, y = 5]$

Is $x = y$? No

Is $x > y$? Yes

$x = x - y = 20 - 5 = 15$ $[x = 15, y = 5]$

Is $x = y$? No

Is $x > y$? Yes

$x = x - y = 15 - 5 = 10$ $[x = 10, y = 5]$

Is $x = y$? No

Is $x > y$? Yes

$x = x - y = 10 - 5 = 5$ $[x = 5, y = 5]$

Is $x = y$? Yes

Output $x = 5$

Flowchart finds HCF of two numbers (Euclid's method).

(3) By performing traces, or otherwise, establish what the following algorithm achieves.

Step 1: Two positive integers are entered.

Step 2: If the two numbers are equal, then output their common value. Otherwise go to Step 3.

Step 3: Divide the larger number by the smaller one (possibly with a remainder). Then go to Step 4.

Step 4: If the division from Step 3 is exact, then output the divisor [ie the number that we are dividing by]. Otherwise go to Step 5.

Step 5: If the division carried out in Step 3 is not exact, then let the divisor and the remainder be the two new numbers, and go to Step 3.

## Solution

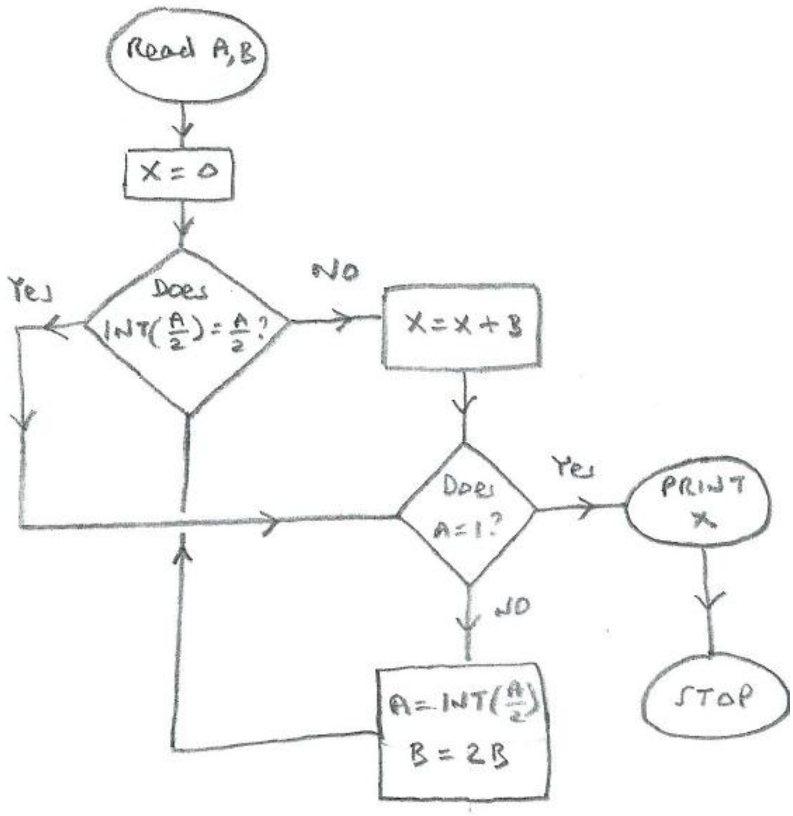
Let $b = ka + r$, where $a$ & $k$ are positive integers, and $r$ is a non-negative integer, with $0 \leq r < a$.

The algorithm uses the result that $hcf(a, b) = hcf(a, r)$.

[See STEP, Pure Exercises, Integers Q7 for proof.]

The pair $(a, b)$ is replaced with the pair $(a, r)$, and the process is then repeated with $(r, a)$, and so on until $r = 0$. (At each stage, $r < a$, and so $r = 0$ after a finite number of steps.

(4) Perform some traces for the following flowchart, in order to establish its purpose.



**Solution**

$A = 4,\ B = 7$

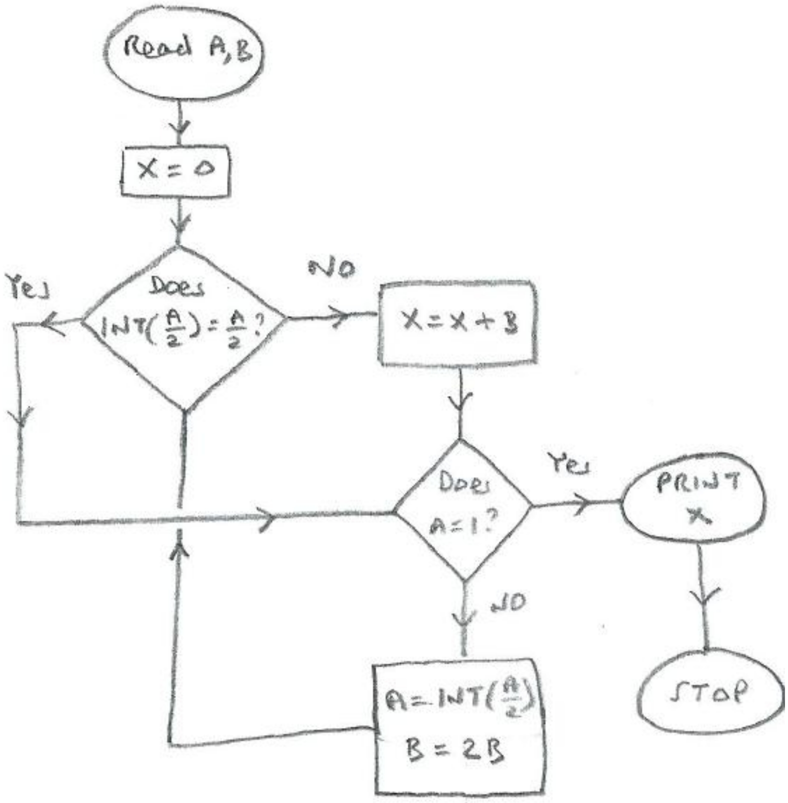
$X = 0$

$A = 2,\ B = 14$

$A = 1,\ B = 28$

$X = 28$

Print 28




$A = 4,\ B = 6$

$X = 0$

$A = 2,\ B = 12$

$A = 1,\ B = 24$

$X = 24$

Print 24

$A = 19,\ B = 6$

$X = 0$

$X = 6$

$A = 9,\ B = 12$

$[19 \times 6 = 6 + 18 \times 6$

$= 6 + 9 \times 12]$

$X = 6 + 12 = 18$

$A = 4,\ B = 24$

$[6 + 9 \times 12$

$= 6 + 12 + 8 \times 12$

$= 18 + 4 \times 24]$

$A = 2, B = 48$

$[18 + 4 \times 24$

$= 18 + 2 \times 48]$

$A = 1, B = 96$

$[18 + 2 \times 48$

$= 18 + 1 \times 96]$

$X = 18 + 96 = 114$

Print 114

The algorithm is multiplying two numbers. It is using the idea that (for the 3rd example):

$8 \times 12 = 4 \times 24 = 2 \times 48 = 1 \times 96$

(it being relatively easy to divide an even number by 2, and to multiply by 2)

When faced with eg $9 \times 12$ (with 9 not being an even number), the algorithm splits this into $12 + (8 \times 12)$.

# Bin Packing

(1) A library needs to store away some of its books (to make way for more computers). Each of the storage boxes can contain 100 books. The shelves to be stored have the following contents, and each shelf has to be stored in a single box.

Photography: 7 books
Fiction: 45 books
Pastimes: 13 books
History: 27 books
Art: 6 books
Computers: 19 books
Biographies: 44 books
Self-help: 15 books
Science : 21 books

(i) Apply the First-fit algorithm.
(ii) Apply the First-fit Decreasing algorithm.
(iii) Apply the Full bins method.

## Solution

(i) Box 1: 7, 45, 13, 27, 6 (98)
Box 2: 19, 44, 15, 21 (99)

(ii) In decreasing order: 45, 44, 27, 21, 19, 15, 13, 7, 6

Box 1: 45, 44, 7 (96)
Box 2: 27, 21, 19, 15, 13 (95)
Box 3: 6

[So the First-fit Decreasing algorithm can, in some cases, be less effective than the First-fit algorithm.]

(iii) For example,
Box 1: 7, 45, 15, 27, 6 (100)
Box 2 : 13, 19, 44, 21 (97)

(2) If the First Fit Decreasing method produces 54 | 3222 | 2 , with bins of size 10, what can be said about the other possible Bin Packing methods?

(2) If the First Fit Decreasing method produces 54 | 3222 | 2 , with bins of size 10, what can be said about the other possible Bin Packing methods?

## Solution

The Full Bins method produces a better solution: 532 | 4222.
Also, the First First method would produce a better solution if the original order was 5324222, for example (or 2352422 etc).

(3) $n$ items are to be packed in bins (all of a certain - unspecified - size) using the First Fit Decreasing algorithm. If the number of comparisons is to be used as a measure of the complexity of the algorithm, determine this complexity in the worst case.

(3) $n$ items are to be packed in bins (all of a certain - unspecified - size) using the First Fit Decreasing algorithm. If the number of comparisons is to be used as a measure of the complexity of the algorithm, determine this complexity in the worst case.

**Solution**

In the worst case, each bin contains only one item (to maximise the number of comparisons).

So total number of comparisons is $1 + 2 + \cdots + (n - 1)$

$= \frac{1}{2}(n - 1)n$

and the order is quadratic; or $O(n^2)$.

# Sorting

(1) Use the Bubble Sort algorithm to sort the following items into increasing order.

7 45 13 27 6 19 44 15 21

(1) Use the Bubble Sort algorithm to sort the following items into increasing order.

7 45 13 27 6 19 44 15 21

**Solution**

Original list: 7 45 13 27 6 19 44 15 21

1st Pass: 7 13 27 6 19 44 15 21 45

2nd Pass: 7 13 6 19 27 15 21 44 45

3rd Pass: 7 6 13 19 15 21 27 44 45

4th Pass: 7 6 13 15 19 21 27 44 45

5th Pass: 7 6 13 15 19 21 27 44 45

[The 5th pass is necessary, in order to establish that no further swaps are needed.]

(2) Use the Quick Sort algorithm to sort the following items into increasing order.

7  45  13  27  6  19  44  15  21

(2) Use the Quick Sort algorithm to sort the following items into increasing order.

7  45  13  27  6  19  44  15  21

**Solution**

Step 1: Select the middle item (or the right-hand of the two middle items) as the pivot [in brackets here, but circled when handwritten]

7  45  13  27  (6)  19  44  15  21

Step 2: Pivot about 6: the remaining items (ie on either side of the 6) that are less than (or equal to) 6 go to the left of (6), without changing their order; similarly, items that are greater than 6 go to the right. The pivot is underlined, to indicate that its position in the list is fixed. Also, the next pivot(s) are labelled.

<u>**6**</u>  7  45  13  27 (19)  44  15  21  [2 marks]

Step 3: The process is repeated until all items are fixed:

<u>**6**</u>  7  (13) 15  <u>**19**</u>  45  27 (44)  21

[Here there are two 'sub-lists': 7  13  15  and  45  27  44  21; both of which produce a pivot.]

<u>**6**</u>  (7)  <u>**13**</u>  (15)  <u>**19**</u>  27 (21)  <u>**44**</u>  (45)

[Strictly speaking, 7, 15 and 45 are pivots, even though they are in sub-lists of 1 item each.]

<u>**6**</u>  <u>**7**</u>  <u>**13**</u>  <u>**15**</u>  <u>**19**</u>  <u>**21**</u>  (27)  <u>**44**</u>  <u>**45**</u>

<u>**6**</u>  <u>**7**</u>  <u>**13**</u>  <u>**15**</u>  <u>**19**</u>  <u>**21**</u>  <u>**27**</u>  <u>**44**</u>  <u>**45**</u>  [2 marks]

Note: Sometimes the first item is chosen as the pivot. Also the method of labelling the pivot varies. It is recommended to describe what is being done, and to define any labelling (so that it is clear that the pivoting process occurs in the line after the pivot has been identified).

(3) A list of $n$ numbers is sorted by making passes through an algorithm.To make a pass, compare the 1st and 2nd numbers. If necessary, swap them so that the 1st number is less than or equal to the 2nd number. Then repeat with the 2nd and 3rd numbers, and so on until the $(n-1)$st and $n$th numbers have been dealt with.

Repeat until a pass occurs with no swaps.

What are the minimum and maximum number of comparisons that are required?

(3) A list of $n$ numbers is sorted by making passes through an algorithm.To make a pass, compare the 1st and 2nd numbers. If necessary, swap them so that the 1st number is less than or equal to the 2nd number. Then repeat with the 2nd and 3rd numbers, and so on until the $(n-1)$st and $n$th numbers have been dealt with.

Repeat until a pass occurs with no swaps.

What are the minimum and maximum number of comparisons that are required?

**Solution**

If the numbers are already in increasing order, then $n-1$ comparisons are required to complete the 1st pass, and the algorithms stops at this point, as there are no swaps.

So $n-1$ is the minimum number of comparisons.

If the largest number is in the 1st position, and the smallest number is in the last position, then $n-1$ passes will be required to move the smallest number to the 1st position, and each of these passes involve $n-1$ comparisons. [Note that, in contrast to the Bubble Sort algorithm, the last number isn't removed after each pass.] Each of these passes will involve a swap, and so a further pass (involving $n-1$ comparisons) will be required. This gives a total of $n(n-1)$ comparisons.

So $n(n-1)$ is the maximum number of comparisons.

**Example 1**: 5231 [Note that the numbers need not be in decreasing order]

[ ] indicates a comparison with a swap

( ) indicates a comparison without a swap

1st pass: [25]31, 2[35]1, 23[15]

2nd pass: (23)15, 2[13]5, 21(35)

3rd pass: [12]35, 1(23)5, 12(35)

4th pass: (12)35, 1(23)5, 12(35)

So $4 \times 3 = 12$ comparisons

**Example 2**: 5213 (where the smallest number is not in the last position)

1st pass: [25]13, 2[15]3, 21[35]

2nd pass: [12]35, 1(23)5, 12(35)

3rd pass: (12)35, 1(23)5, 12(35)