

Algorithms and Flowcharts

1 An algorithm is shown below.

- 1 Input n
- 2 Let $m = 4n - 3$
- 3 Print m

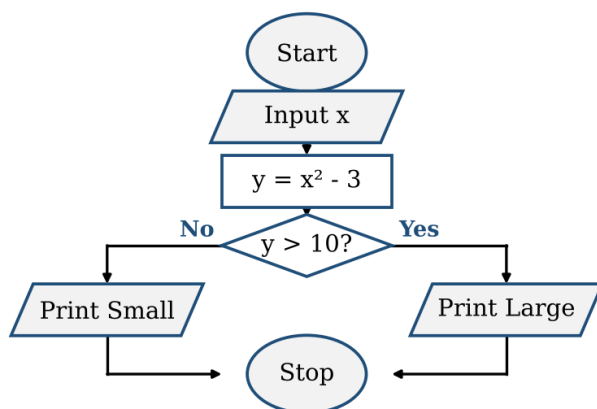
- (a) Write down the output when $n = 5$. [1]
- (b) Write down the output when $n = -2$. [1]

2 An algorithm is shown below.

- 1 Input a and b
- 2 Let $c = a + b$
- 3 Let $d = c \div 2$
- 4 Print d

- (a) Find the output when $a = 8$ and $b = 14$. [1]
- (b) State what the value of d represents. [1]

3 The flowchart shows an algorithm.



- (a) Find the output when $x = 5$. [1]
- (b) Find the output when $x = 2$. [1]

4 An algorithm is shown below.

- 1 Input a
- 2 If a is even, Let $r = a \div 2$
- 3 If a is odd, Let $r = 3a + 1$
- 4 Print r

- (a) Find the output when $a = 14$. [1]
- (b) Find the output when $a = 7$. [1]

5 An algorithm is shown below.

- 1 Input a and b
- 2 Let $q = \text{INT}(a \div b)$
- 3 Let $r = a - b \times q$
- 4 Print r

$\text{INT}(x)$ means the integer part of x . Find the output when $a = 47$ and $b = 6$, and state what r represents. [3]

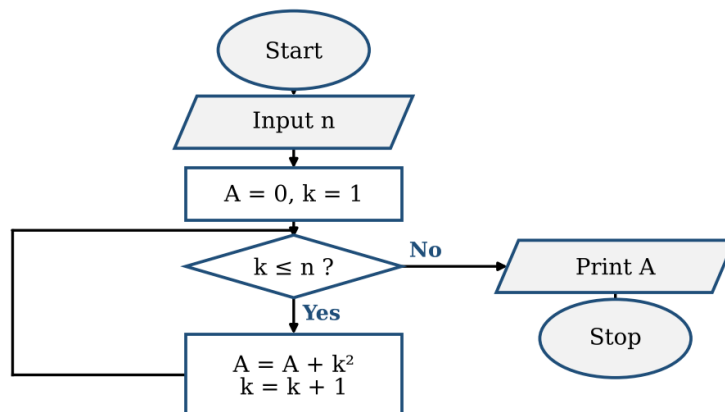
6 An algorithm is shown below, with the list 7, 12, 3, 9, 15.

- 1 Let M be the first number in the list
- 2 Look at each remaining number in turn. If it is larger than M , replace M by it
- 3 Print M

Find the output and state what the algorithm does. [2]

- 7 An algorithm is shown below.
- 1 Let $x = 1$ and $t = 0$
 - 2 Let $t = t + x$
 - 3 Let $x = x + 2$
 - 4 If $x > 9$ go to step 5, otherwise go to step 2
 - 5 Print t
- (a) Find the output. [3]
- (b) State what the algorithm calculates. [1]

- 8 The flowchart shows an algorithm. Apply it with $n = 4$, showing the value of A each time the test is made, and write down the output.



- 9 An algorithm is shown below.
- 1 Input a and b , where $a > b$
 - 2 Let r be the remainder when a is divided by b
 - 3 If $r = 0$, Print b and stop
 - 4 Let $a = b$ and $b = r$, then go to step 2
- Apply the algorithm with $a = 84$ and $b = 36$, and state what the output represents. [4]

- 10 An algorithm is shown below.
- 1 Input a positive integer n
 - 2 Let r be the remainder when n is divided by 2 and write r down
 - 3 Let $n = \text{INT}(n \div 2)$
 - 4 If $n > 0$ go to step 2, otherwise stop
 - 5 Read the digits written down in reverse order
- Apply the algorithm with $n = 45$ and state what the final answer represents. [4]

- 11 An algorithm is shown below.
- 1 Input n ; Let $a = 1$, $b = 1$ and $k = 2$
 - 2 Let $c = a + b$
 - 3 Let $a = b$ and Let $b = c$
 - 4 Let $k = k + 1$
 - 5 If $k < n$ go to step 2, otherwise Print b
- (a) Find the output when $n = 6$, giving the value of b after each pass. [3]
- (b) The numbers 1, 1, 2, 3, 5, 8, ... form a sequence. State the name of this sequence and say which term of it is printed when $n = 6$. [1]

- 12** The algorithm below is meant to print the sum of the integers from 1 to n .

1 Input n
 2 Let $T = 1$ and $k = 1$
 3 Let $T = T + k$
 4 Let $k = k + 1$
 5 If $k \leq n$ go to step 3, otherwise Print T

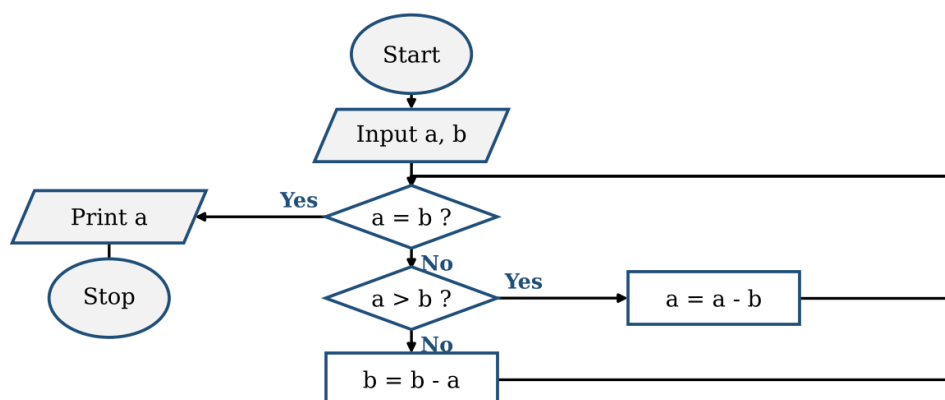
(a) Find the output when $n = 3$. [2]

(b) Identify the error in the algorithm and state how to correct it. [2]

- 13** An algorithm is shown below.

1 For each integer k from 1 to 12
 2 If k is a multiple of 12, print AB
 3 Otherwise if k is a multiple of 3, print A
 4 Otherwise if k is a multiple of 4, print B
 5 Otherwise print k
 Write down the output. [3]

- 14** The flowchart shows an algorithm.



(a) Show the successive values of a and b when the algorithm is applied with $a = 48$ and $b = 18$, and write down the output. [4]

(b) State what the output represents. [1]

- 15** An algorithm is shown below.

1 Input an integer n , $n \geq 2$
 2 Let $d = 2$
 3 If $d \times d > n$, Print "Prime" and stop
 4 If n is divisible by d , Print "Not prime" and stop
 5 Let $d = d + 1$ and go to step 3

(a) Apply the algorithm with $n = 91$. [2]

(b) Apply the algorithm with $n = 97$ and state the final value of d . [3]

(c) Explain why it is enough to test values of d with $d \times d \leq n$. [1]

- 16** Write an algorithm, in numbered steps, that inputs a list of n numbers and prints their mean. [4]

- 17** An algorithm is shown below.

1 Input N
 2 Let $x = 1$ and $k = 0$
 3 Let $x = (x + N \div x) \div 2$ and Let $k = k + 1$
 4 If $k < 3$ go to step 3
 5 Print x

Apply the algorithm with $N = 10$, giving the value of x after each pass to 4 decimal places, and state the output to 3 decimal places. Compare it with $\sqrt{10}$. [4]

18 An algorithm is shown below.

1 Input a positive integer n

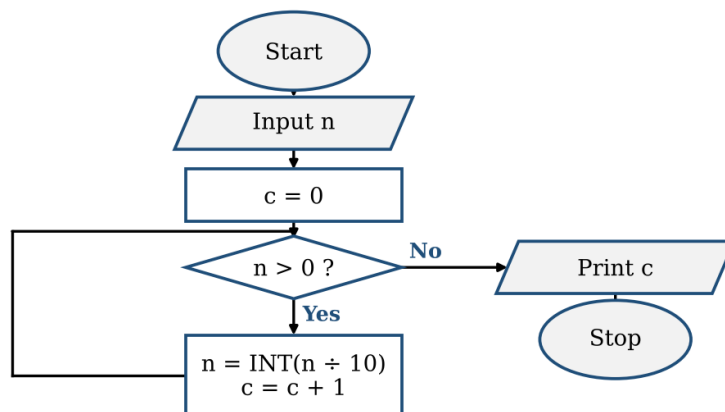
2 Print n

3 If $n < 10$, stop

4 Let n be the sum of the digits of n and go to step 2

Find the output when $n = 98765$ and describe what happens when the input is a multiple of 9. [4]

19 The flowchart shows an algorithm. $\text{INT}(x)$ is the integer part of x .



(a) Find the output when $n = 3072$, showing the values of n and c after each pass. [3]

(b) The algorithm is supposed to print the number of digits of n . State an input for which it gives the wrong output, and the output given. [2]

20 An algorithm is shown below.

1 Input three positive numbers and let c be the largest and a, b the other two

2 If $a + b \leq c$, Print "Not a triangle" and stop

3 If $a^2 + b^2 = c^2$, Print "Right-angled" and stop

4 Print "Other"

Write down the output for each of the inputs

(a) 5, 12, 13 [1]

(b) 6, 8, 11 [2]

(c) 3, 4, 8 [1]

21 The following algorithm estimates the area under the graph of $y = 1/x$ between $x = A$ and $x = B$.

1 Input A, B and N

2 Let $H = (B - A) \div N$

3 Let $S = 1/A + 1/B$

4 Let $X = A$

5 Let $X = X + H$

6 If $X \geq B$ go to step 9

7 Let $S = S + 2/X$

8 Go to step 5

9 Let $T = H \times S \div 2$

10 Print T

(a) Apply the algorithm with $A = 1, B = 3$ and $N = 4$, showing the values of X and S each time step 6 is reached, and state the output as a fraction. [4]

(b) Find the percentage error in the output as an estimate of the exact area $\ln 3$, giving your answer to 3 significant figures, and say what the algorithm is doing. [3]

- 22** An algorithm is shown below.
 1 Input a positive integer N and Let $s = 0$
 2 If $(s + 1)^2 > N$, Print s and stop
 3 Let $s = s + 1$ and go to step 2
 (a) Apply the algorithm with $N = 50$, stating the successive values of s . [3]
 (b) Show that the algorithm outputs the largest integer s such that $s^2 \leq N$. [3]
 (c) State, in terms of N , the number of times step 3 is carried out and explain why this algorithm is inefficient for large N . [2]
-
- 23** The list 3, 8, 12, 17, 20, 25 is stored in positions 1 to 6. An algorithm is shown below.
 1 Input a target T ; Let $i = 1$ and $j = 6$
 2 If $i \geq j$, Print "None" and stop
 3 Let s be the sum of the numbers in positions i and j
 4 If $s = T$, Print the two numbers and stop
 5 If $s < T$, Let $i = i + 1$, otherwise Let $j = j - 1$; go to step 2
 (a) Apply the algorithm with $T = 37$, showing i , j and s at each pass. [4]
 (b) Apply the algorithm with $T = 30$ and state the output. [3]
 (c) Explain why, when $s < T$ in step 5, the number in position i cannot be part of a pair with sum T . [2]
-
- 24** An algorithm is shown below.
 1 Input a and n ; Let $r = 1$
 2 If $n = 0$, Print r and stop
 3 If n is odd, Let $r = r \times a$
 4 Let $a = a \times a$ and Let $n = \text{INT}(n \div 2)$
 5 Go to step 2
 (a) Apply the algorithm with $a = 3$ and $n = 13$, showing the values of a , n and r at the start of each pass. [4]
 (b) State what is calculated and write 13 in binary to explain which passes multiply r . [2]
-
- 25** A book code has nine digits $d_1, d_2, \dots, d_9$ followed by a check digit c . An algorithm to find c is shown below.
 1 Let $S = 10d_1 + 9d_2 + 8d_3 + \dots + 2d_9$
 2 Let c be the number from 0 to 10 such that $S + c$ is a multiple of 11 (write X if $c = 10$)
 3 Print c
 (a) Find the output when the digits are 0, 3, 0, 6, 4, 0, 6, 1, 5. [3]
 (b) A student swaps the digits d_2 and d_3 (here 3 and 0). Find the new check digit, and explain why this type of code always detects the swap of two adjacent different digits. [4]
-
- 26** An algorithm is shown below.
 1 Input an integer $n > 1$; Let $s = 0$
 2 For each integer d from 1 to $n - 1$: if n is divisible by d , Let $s = s + d$
 3 Print s
 4 If $s = n$, Print "Perfect"
 (a) Find the output when $n = 28$ and when $n = 12$. [3]
 (b) How many times is the test in step 2 made when $n = 496$? [1]
 (c) A student proposes running d from 1 to $\text{INT}(n \div 2)$ only. Explain why the output is unchanged, and find the saving in the number of tests when $n = 496$. [3]

- 27** An algorithm is shown below.
 1 Input a positive integer n and Let $c = 0$
 2 If $n = 1$, Print c and stop
 3 If n is even, Let $n = n \div 2$, otherwise Let $n = 3n + 1$
 4 Let $c = c + 1$ and go to step 2
 (a) Show that the only input for which the output is 4 is $n = 16$. [3]
 (b) Find all the inputs for which the output is 8, justifying your answer by working backwards from 1. [6]
-
- 28** An algorithm is shown below.
 1 Input positive integers a and b with $a \geq b$
 2 Let r be the remainder when a is divided by b
 3 If $r = 0$, Print b and stop
 4 Let $a = b$ and Let $b = r$ and go to step 2
 (a) Apply the algorithm with $a = 1071$ and $b = 462$. [3]
 (b) Show that if $a = qb + r$ then every common divisor of a and b is a common divisor of b and r , and vice versa. Hence explain why the algorithm outputs the highest common factor of a and b . [4]
 (c) Explain why the algorithm always terminates. [2]
-
- 29** An algorithm is shown below.
 1 Input a positive integer N ; Let $n = 0$ and $s = 0$
 2 Let $n = n + 1$ and Let $s = s + n$
 3 If $s > N$, Print n and stop, otherwise go to step 2
 (a) Show that the output is the smallest positive integer n such that $n(n + 1) > 2N$. [3]
 (b) Find the output when $N = 500$, and when $N = 10^6$. [4]
 (c) Show that the output is approximately $\sqrt{2N}$ for large N , and write down a quicker way to find it than running the loop. [3]
-
- 30** An algorithm is shown below.
 1 Input a positive integer n and Let $c = 0$
 2 Let R be the number formed by writing the digits of n in reverse order
 3 If $R = n$, Print n and c and stop
 4 Let $n = n + R$ and Let $c = c + 1$; go to step 2
 (a) Find the output when $n = 87$. [3]
 (b) Find the output when $n = 120$, and explain what is unusual about the first reversal. [3]
 (c) Show that the algorithm stops after one addition for every two-digit number that is not a palindrome and whose digits sum to at most 9. Explain, with an example of digit sum 10, why this fails if the digits sum to more than 9. [4]

Sorting Algorithms

- 1** Apply the first pass of a bubble sort to put the list 8, 3, 5, 9, 2, 6 into ascending order. Write down the list at the end of the pass. [2]
-
- 2** A quick sort into ascending order is applied to the list 14, 9, 20, 5, 11, 17, 3.
 Use the middle item as the pivot (for a list with an even number of items, the right-hand of the two middle items), place items equal to the pivot to the right, and treat a sub-list of one item as sorted.
 State the pivot and write down the list after the first pass. [2]
-
- 3** Use the first-fit algorithm to pack the items 4, 7, 3, 5, 2, 6 into bins of size 10. [2]

- 4** The numbers 4, 9, 13, 18, 23, 31, 40 are in ascending order. Use a binary search to locate 31. Use the middle item (for an even number of items, the right-hand of the two middle items) and discard that item and the half that cannot contain the target. List the items you use as the middle item at each stage. [2]
-
- 5** A bubble sort is applied to a list of 9 numbers.
 (a) How many comparisons are made in the first pass? [1]
 (b) What is the greatest number of passes that could be needed, if a final pass with no swaps is not counted? [1]
-
- 6** A list of numbers is to be sorted into descending order using a bubble sort, starting at the left-hand end of the list.
 (a) Describe how to carry out the first pass. [2]
 (b) Apply the first pass to the list 5, 8, 3, 9, 4 and write down the list at the end of the pass. [2]
-
- 7** Use a bubble sort to put the list 12, 5, 9, 3, 8 into ascending order, showing the list after each pass. State the total number of swaps made. [5]
-
- 8** Use a bubble sort to put the list 4, 9, 2, 7, 6, 3 into descending order, showing the list after each pass. [4]
-
- 9** Use a quick sort to put the list 23, 8, 15, 4, 19, 11, 30 into ascending order. Use the middle item as the pivot (for a list with an even number of items, the right-hand of the two middle items), place items equal to the pivot to the right, and treat a sub-list of one item as sorted. Show the list after each pass and underline or circle the pivots. [4]
-
- 10** Seven items have sizes 6, 4, 5, 3, 7, 2, 5. They are to be packed into bins of size 12.
 (a) Use the first-fit algorithm to pack the items. [2]
 (b) Find a lower bound for the number of bins and state whether your packing is optimal. [2]
-
- 11** Eight items have sizes 9, 4, 6, 2, 8, 3, 5, 7. Use the first-fit decreasing algorithm to pack them into bins of size 12. [3]
-
- 12** The list 3, 7, 12, 15, 21, 24, 28, 33, 40, 47 is in ascending order. Use a binary search to show that 26 is not in the list. Use the middle item (for an even number of items, the right-hand of the two middle items) and discard that item and the half that cannot contain the target. [3]
-
- 13** Eight items have sizes 5, 7, 4, 3, 6, 5, 2, 8 and bins have size 10.
 (a) Use the first-fit algorithm. [2]
 (b) Find a packing that uses the minimum number of bins by first pairing items that make a full bin, and explain why it is optimal. [3]
-
- 14** The list 17, 3, 11, 25, 6, 14, 9, 20 is sorted into ascending order using a bubble sort. The sort stops after a pass in which no swaps are made.
 (a) Show the list after each pass. [3]
 (b) State the total number of comparisons and the total number of swaps made. [2]
-
- 15** Use a quick sort to put the list 18, 7, 25, 12, 30, 4, 21, 15, 9 into descending order. Use the middle item as the pivot (for a list with an even number of items, the right-hand of the two middle items), place items equal to the pivot to the right, and treat a sub-list of one item as sorted. (In descending order the items larger than the pivot go to the left.) [5]
-
- 16** Six students are to be listed in alphabetical order by first name using a bubble sort: Kim, Abi, Zoe, Tom, Eve, Ben. Show the order after each pass and state how many passes are needed before the sort stops. [4]
-

- 17** A builder needs pipes of lengths 1.8, 2.4, 3.1, 0.9, 2.2, 1.5, 2.7, 1.2 metres, each cut from stock lengths of 5 m.
- Use the first-fit algorithm to find the number of stock lengths needed. [2]
 - Use first-fit decreasing and state whether it improves on part (a). [3]
 - Show that no packing can use fewer stock lengths than your answer to (b). [2]
-
- 18** The list 5, 9, 14, 20, 26, 31, 37, 44, 52, 58, 63, 71 is in ascending order. Use a binary search to find 14. Use the middle item (for an even number of items, the right-hand of the two middle items) and discard that item and the half that cannot contain the target. Write down the middle item at each stage and state how many items are examined. [5]
-
- 19** A bubble sort is used to put a list of numbers into ascending order. Explain why, after the first pass, the largest number is in the last position, whatever the original order. [3]
-
- 20** Ten items have sizes 5, 4, 6, 8, 3, 3, 9, 7, 3, 2. Bins have size 12.
- Use first fit to pack the items and find the total unused space in the bins. [3]
 - Explain why the packing in part (a) cannot be improved. [2]
-
- 21** A student applies a bubble sort (ascending) to the list 9, 4, 7, 2, 8. After the first pass the student writes 4, 7, 9, 2, 8.
- Identify the mistake the student has made. [2]
 - Write down the correct list after pass 1 and after pass 2. [2]
 - State how many passes in total are needed for this list. [1]
-
- 22** A quick sort into descending order is applied to the list 14, 6, 14, 9, 2, 14, 11, 6, 9, 20, 3, which contains repeated values. Use the middle item as the pivot (for a list with an even number of items, the right-hand of the two middle items), place items equal to the pivot to the right, and treat a sub-list of one item as sorted. (In descending order the items larger than the pivot go to the left.) Show every pass and state how many passes are needed. [5]
-
- 23** Eight items are to be packed into bins of size 20. In the order given, the sizes are 9, 6, 12, x , 8, 5, 7, 11, where x is an integer different from the other sizes. The first-fit algorithm gives this packing.
Bin 1: 9, 6, 5 Bin 2: 12, 8 Bin 3: x Bin 4: 7, 11
- Explain why $13 < x \leq 20$. [3]
 - The first-fit decreasing algorithm also uses four bins, with x in a bin on its own, and the total unused space in the four bins is 5. Find the value of x . [3]
-
- 24** A sorted list contains 100 names and a binary search is used to find a name. Use the middle item (for an even number of items, the right-hand of the two middle items) and discard that item and the half that cannot contain the target.
- Show that at most 7 names are examined. [2]
 - Find, with justification, the greatest list length for which at most 7 names are examined. [2]
 - A different list has 1000 names. State the greatest number of examinations needed. [2]
-
- 25** A carpenter must cut pieces of lengths 1.2, 0.8, 2.0, 1.5, 0.6, 1.1, 0.9, 1.4, 1.0, 0.5 metres from planks of length 3.0 m. A saw cut wastes no wood.
- Use first fit to find the number of planks required. [2]
 - Use first-fit decreasing to find the number of planks required. [2]
 - Show that the plan in part (b) uses the fewest possible planks, and find the total length of wood wasted. [3]

- 26** A bubble sort into ascending order is applied to the list 6, 2, 7, 1, 5, 3, 4.
- (a) Show the list after each pass and the number of swaps in each pass. [3]
 - (b) Show that the total number of swaps is 12, and explain why each swap removes exactly one pair of items that are in the wrong order. [2]
 - (c) Write down the number of swaps needed for the list 7, 6, 5, 4, 3, 2, 1 and explain your answer. [2]
-
- 27** A bubble sort (ascending) is applied to a list of n different numbers. A pass is made over the whole of the unsorted part of the list, which shortens by one after each pass.
- (a) Show that the greatest number of comparisons in the sort (without a final pass with no swaps) is $\frac{n(n-1)}{2}$. [2]
 - (b) Show that the number of swaps equals the number of pairs of numbers that are in the wrong order, and hence state the list orderings for which the number of swaps is greatest. [3]
 - (c) For $n = 50$ in the reverse order, find the number of swaps and the number of comparisons, and comment on your answers. [4]
-
- 28** Quick sort is applied to lists of the numbers 1 to 7 in some order. Use the middle item as the pivot (for a list with an even number of items, the right-hand of the two middle items), place items equal to the pivot to the right, and treat a sub-list of one item as sorted.
- (a) Show that for the list 5, 3, 6, 4, 7, 2, 1 the sort is completed in 2 passes. [3]
 - (b) Show that a list of the numbers 1 to 7 can be sorted in 2 passes only if the first pivot is 4. [3]
 - (c) Show that the list 1, 3, 5, 7, 6, 4, 2 needs 6 passes, and explain why 6 is the greatest number of passes possible for 7 items. [2]
-
- 29** Eight items have sizes 7, 7, 3, 4, 7, 5, 6, 5 and bins have size 15.
- (a) Use first-fit decreasing to pack the items. [2]
 - (b) Find a packing that uses only 3 bins. [2]
 - (c) Hence comment on the first-fit decreasing algorithm, and explain why 3 is the minimum number of bins. [2]
-
- 30** A binary search is used on an ordered list of 15 different numbers, to locate a number that is known to be in the list. Use the middle item (for an even number of items, the right-hand of the two middle items) and discard that item and the half that cannot contain the target.
- (a) Show that exactly one number is found at the first examination, two numbers at the second, four at the third and eight at the fourth. [4]
 - (b) Find the mean number of examinations when each of the 15 numbers is equally likely to be the target. [3]
 - (c) Write down a formula for the greatest number of examinations needed for a list of $2^k - 1$ numbers and use it to find the greatest number for 1023 numbers. [3]

Algorithm Efficiency

- 1** A bubble sort is used on a list of 10 numbers. Find the greatest number of comparisons that could be made if every pass is made in full and the passes shorten by one each time. [2]
-
- 2** The time taken by a bubble sort is proportional to n^2 , where n is the number of items in the list. By what factor does the time increase when the length of the list is doubled? [2]
-
- 3** A bubble sort takes 0.8 seconds to sort 500 numbers. Assuming that the time is proportional to n^2 , estimate the time to sort 1000 numbers. [2]

- 4** A salesperson must visit 5 towns, starting and ending at the same town, and visiting each other town exactly once. All pairs of towns are joined by a road. Find the number of different possible tours, where a tour and its reverse count as the same. [2]
- 5** A sorted list has 63 items. Show that a binary search never needs to examine more than 6 items. [2]
- 6** Find the value of $\frac{n(n-1)}{2}$ when $n = 20$, and state what it represents for a bubble sort of 20 items. [2]
- 7** The time taken by a quick sort is proportional to $n \log n$, where n is the number of items in the list. A quick sort takes 0.45 seconds to sort 4000 numbers. Estimate the time to sort 60 000 numbers. [3]
- 8** A bubble sort is applied to lists of different lengths. The greatest number of comparisons for a list of n items is $\frac{n(n-1)}{2}$.
 (a) Find the greatest number of comparisons for lists of 30 items and 60 items. [2]
 (b) Find the ratio of the two answers, giving it to 3 significant figures, and explain why it is close to 4. [2]
- 9** A travelling salesman problem has 8 towns and every pair of towns is joined by a road. A computer checks every possible tour, counting a tour and its reverse as the same, at a rate of one million tours per second. Find the number of tours and the time needed. [3]
- 10** An algorithm for problem A takes n^2 steps for an input of size n , and an algorithm for problem B takes 2^n steps.
 (a) Complete the comparison by finding the number of steps for each algorithm when $n = 4$, 5 and 10. [2]
 (b) Write down the smallest integer $n \geq 3$ for which 2^n is greater than n^2 , and describe what happens for larger n . [2]
- 11** An algorithm takes $\frac{n(n+1)}{2}$ steps for an input of size n . Find the number of steps when $n = 100$ and when $n = 200$, and the ratio of the second to the first. [3]
- 12** A list of 1000 sorted items is searched by a binary search, and also by a linear search (checking the items in order from the first). Find the greatest number of items examined by each method. [3]
- 13** The time for an algorithm is modelled by $T = 3 \times 10^{-6} n^2$ seconds. Find the time for $n = 2 \times 10^4$, giving your answer in minutes to 3 significant figures. [3]
- 14** A bubble sort takes 3.2 seconds on a list of 4000 numbers. Assume the time is proportional to n^2 .
 (a) Estimate the time for a list of 7000 numbers. [2]
 (b) Estimate the greatest length of list that can be sorted in one minute. [3]
- 15** Two algorithms solve the same problem for an input of size n . Algorithm A takes $0.01 n^2$ milliseconds and algorithm B takes $100 n$ milliseconds.
 (a) Find the value of n for which the two algorithms take the same time. [2]
 (b) State, with a reason, which algorithm is faster for $n = 5000$ and which for $n = 50\,000$. [2]
 (c) Explain why the algorithm with the smaller multiplier is not always the better choice. [1]

- 16** An algorithm compares every pair of items in a list of n items exactly once, using these steps.
 1 For each i from 1 to $n - 1$
 2 For each j from $i + 1$ to n , compare item i with item j
 (a) Show that the number of comparisons is $\frac{n(n-1)}{2}$. [2]
 (b) A computer performs 2×10^6 comparisons per second. Find the time to compare all pairs in a list of 3000 items. [2]
-
- 17** A computer checks one hundred million (10^8) different tours per second in a travelling salesman problem with n cities in which every pair of cities is joined directly. A tour and its reverse count as the same tour.
 (a) Find the time to check all tours when $n = 15$. [3]
 (b) Find the time to check all tours when $n = 20$, giving your answer in years (take 1 year = 3.15×10^7 seconds). [3]
-
- 18** The table shows the time taken by an algorithm for inputs of size n .
 n : 100, 200, 400
 t (seconds): 0.05, 0.21, 0.79
 (a) Show that the times are consistent with a time proportional to n^2 . [2]
 (b) Use the result for $n = 400$ to estimate the time for $n = 1000$. [2]
-
- 19** For a list of n items, bubble sort needs up to $\frac{n(n-1)}{2}$ comparisons, while a good quick sort needs about $n \log_2 n$ comparisons.
 (a) Find both numbers for $n = 1024$. [2]
 (b) Find how many times as many comparisons bubble sort needs, to 3 significant figures. [2]
-
- 20** The first-fit algorithm packs n items one at a time. Each item is compared with the bins in order until it fits.
 (a) Find the greatest number of size comparisons made when the n items need n bins (each item is too large to share a bin). [2]
 (b) Hence state the order of the algorithm in the worst case and find the number of comparisons when $n = 200$. [2]
-
- 21** Algorithm P takes $1000n^2$ steps and algorithm Q takes 2^n steps for an input of size n .
 (a) Show that for $n = 10$, algorithm Q needs fewer steps than P. [2]
 (b) Find the smallest n for which algorithm Q needs more steps than P, showing the values at $n = 18$ and $n = 19$. [3]
 (c) Explain what this shows about choosing between algorithms for large inputs. [1]
-
- 22** A bubble sort on a certain computer takes t seconds to sort n numbers, where $t = 2.5 \times 10^{-7} n^2$.
 (a) Find the time to sort 20 000 numbers. [2]
 (b) Find the greatest number of items that can be sorted in 10 minutes. [3]
 (c) The computer is replaced by one that is 25 times as fast. By what factor does the greatest list that can be sorted in 10 minutes increase? Explain your answer. [3]
-
- 23** A sorted list of n items is searched for a target. A linear search examines the items in order and, on average, finds a present target after examining $\frac{n+1}{2}$ items. A binary search examines at most k items where $2^{k-1} \leq n < 2^k$.
 (a) For $n = 1\,000\,000$, find the average number of items examined by the linear search. [2]
 (b) For $n = 1\,000\,000$, find the greatest number of items examined by the binary search. [2]
 (c) A list has 10^9 items. Find the greatest number of items examined by binary search, and comment on the work saved compared with the average for a linear search. [3]

- 24** In the route inspection (Chinese postman) problem with $2k$ odd vertices, the odd vertices must be paired up. The number of different pairings is $\frac{(2k)!}{2^k k!}$.
- Find the number of pairings when there are 4 odd vertices, and when there are 6 odd vertices. [2]
 - Find the number of pairings for 10 odd vertices and for 12 odd vertices. [3]
 - Show that going from $2k$ to $2k + 2$ odd vertices multiplies the number of pairings by $2k + 1$, and explain why checking all pairings is impractical for large networks. [3]
-
- 25** The time, in milliseconds, taken by an algorithm for an input of size n is modelled by $T = 0.002n^2 + 0.5n + 20$.
- Find T for $n = 100$ and for $n = 10\,000$. [2]
 - Find the percentage of the time for $n = 10\,000$ that comes from the n^2 term, to 3 significant figures. [2]
 - Find the smallest integer n for which the n^2 term is larger than the other two terms combined. [4]
-
- 26** A quick sort is applied to a list of 7 different numbers, and at every stage the pivot happens to be the median of its sub-list. In each pass every item in a sub-list is compared once with that sub-list's pivot, so a sub-list of m items needs $m - 1$ comparisons.
- Show that the total number of comparisons is 10. [3]
 - Find the total for 15 items in the same best case. [3]
 - Compare with the greatest number of comparisons in a bubble sort of 15 items. [2]
-
- 27** A quick sort is applied to a list of $n = 2^k - 1$ different numbers, and at every stage the pivot is the median of its sub-list. A sub-list of m items needs $m - 1$ comparisons in its pass.
- Show that in pass j ($j = 1, 2, \dots, k - 1$) there are 2^{j-1} sub-lists, each of $2^{k-j+1} - 1$ items, and hence that the number of comparisons in pass j is $2^{j-1}(2^{k-j+1} - 2)$. [3]
 - Show that the total number of comparisons is $k \times 2^k - 2(2^k - 1)$, that is $(n + 1)k - 2n$. [3]
 - Find the total for $n = 1023$ and compare with the greatest number for a bubble sort of 1023 items. [3]
-
- 28** A computer checks 10^9 tours per second in a travelling salesman problem with n cities in which every pair of cities is joined directly. A tour and its reverse are the same tour, so there are $\frac{(n-1)!}{2}$ tours.
- Find the greatest number of cities for which all the tours can be checked in one hour. [4]
 - A computer 1000 times as fast is used. Find how many more cities can be handled in one hour, and explain why the improvement is so small. [4]
-
- 29** An algorithm examines every set of three different items in a list of n items exactly once.
- Show that the number of sets examined is $\frac{n(n-1)(n-2)}{6}$. [2]
 - Find the number of sets for $n = 200$, and the ratio of the number for $n = 200$ to the number for $n = 100$, to 3 significant figures. [3]
 - Explain why the ratio is close to 8, and state the order of the algorithm. [2]
 - On a computer the algorithm takes 0.4 seconds for $n = 100$. Estimate the time for $n = 500$. [2]

30 A student makes these claims about a bubble sort of n items ($n > 1$) in which each pass runs over the unsorted part of the list and the sort stops after a pass with no swaps.

- (a) Claim 1: "If the list is already in order the sort makes the same number of comparisons as for a list in reverse order." Show that this is false for $n = 6$. [2]
 - (b) Claim 2: "Doubling the number of items doubles the time." Show that this is false for the worst case, using $n = 50$ and $n = 100$. [2]
 - (c) Claim 3: "Algorithm A takes $0.01 n^2$ ms and algorithm B takes $10n$ ms, so A, with its smaller multiplier, is always faster." Find the value of n at which they take equal times and say which is faster beyond it. [3]
 - (d) Write down the best-case number of comparisons for n items and the worst-case number, and hence find the ratio of the time for a list in reverse order to the time for a list already in order when $n = 1000$. [3]
-