

2.3.2

**ALGORITHMS - SORTING
& SEARCHING & BIG O
TOPIC WISE EXAM QUESTIONS**

A-LEVEL

OCR

(iii) State the purpose of the variable `holdValue` in the procedure `bubbleSort`.

.....
.....
.....
.....
.....
..... [3]

(iv) The procedure `bubbleSort` will only partially sort the array `numbers` into order.

Describe what the programmer would need to add to the algorithm to enable it to fully sort the numbers into order.

.....
.....
.....
..... [2]

(c) (i) The array `numbers` contains 356 numbers to be sorted by the bubble sort algorithm.

State the maximum number of passes a bubble sort would need to complete to sort 356 numbers into order.

.....
..... [1]

(ii) State the name of **one** other sorting algorithm.

.....
..... [1]

- 1 A computer program stores data in an array named `words`.
- (a) The data in the array needs to be searched for a value that the user inputs.
- (i) One example of a searching algorithm is a binary search.

Identify the precondition for a binary search.

.....
..... [1]

- (ii) A second example of a searching algorithm is a linear search.

Describe how a linear search works.

.....
.....
.....
.....
.....
.....
.....
.....
..... [4]

(b) Trudi has written a procedure, bubbleSort.

```
01 procedure bubbleSort (numbers)
02     do
03         sorted = true
04         for count = 0 to numbers.length -2
05             if numbers[count] > numbers[count+1] then
06                 temp = numbers[count+1]
07                 numbers[count+1] = numbers[count]
08                 numbers[count] = temp
09                 sorted = false
10             endif
11         next count
12     until sorted == true
13 endprocedure
```

(i) Identify a line in the procedure bubbleSort where a decision is taken.

.....
..... [1]

(ii) Identify the name of the parameter used in the procedure bubbleSort.

.....
..... [1]

(iii) Describe the purpose of the temp variable in the procedure bubbleSort.

.....
.....
.....
..... [2]

(iv) Describe the purpose of the sorted variable in the procedure bubbleSort.

.....
.....
.....
..... [2]

2 The pseudocode function `binarySearch()` performs a binary search on the array `dataArray` that is passed as a parameter. The function returns the array index of `searchValue` within the array, and `-1` if it is not in the array.

(a) The pseudocode binary search algorithm is incomplete.

(i) Complete the algorithm by filling in the missing statements.

```
function binarySearch(dataArray:byref, upperbound, lowerbound, .....)  
  while true  
    middle = lowerbound + ((upperbound - lowerbound) .....)  
    if upperbound < lowerbound then  
      return .....  
    else  
      if dataArray[middle] < searchValue then  
        lowerbound = .....  
      elseif dataArray[middle] > searchValue then  
        upperbound = .....  
      else  
        return .....  
      endif  
    endif  
  endwhile  
endfunction
```

[6]

(ii) The algorithm uses a while loop.

State a different type of loop that could be used instead of the while loop in the given algorithm.

.....
..... [1]

- (b) The tables below show possible Big O complexities for the worst-case space, best-case space and average time for search algorithms.

Tick the worst-case space complexity for a binary and linear search.

	Binary search	Linear search
$O(\log(n))$		
$O(1)$		
$O(n)$		

Tick the best-case space complexity for a binary and linear search.

	Binary search	Linear search
$O(\log(n))$		
$O(1)$		
$O(n)$		

Tick the average time complexity for a binary and linear search.

	Binary search	Linear search
$O(\log(n))$		
$O(1)$		
$O(n)$		

[6]

- (c) Identify **one** situation where a linear search is more appropriate than a binary search.

.....

..... [1]

4 The following pseudocode procedure performs an insertion sort on the array parameter.

```
01 procedure insertionSort (dataArray:byRef)
02   for i = 1 to dataArray.Length - 1
03     temp = dataArray[i]
04     tempPos = i - 1
05     exit = false
06     while tempPos >= 0 and exit == false
07       if dataArray[tempPos] < temp then
08         dataArray[tempPos + 1] = dataArray[tempPos]
09         tempPos = tempPos - 1
10     else
11       exit = true
12     endif
13   endwhile
14   dataArray[tempPos + 1] = temp
15 next i
16 endprocedure
```

(a) Explain why `dataArray` is passed by reference and not by value.

.....
.....
.....
..... [2]

(b) State whether the procedure `insertionSort` sorts the data into ascending or descending order and explain your choice.

.....
.....
.....
.....
.....
..... [3]

2 Poppy would like to use a bubble sort to sort 250 000 numbers into order from lowest to highest.

Currently the first five numbers before they have been sorted are:

195 584	167 147	158 187	160 125	184 236
---------	---------	---------	---------	---------

Currently the last five numbers before they have been sorted are:

1058	19558	1915	20215	15
------	-------	------	-------	----

(a)* Discuss how a bubble sort works and how efficient it will be when sorting these 250 000 items into order from lowest to highest. [9]

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

(b) State the number of comparisons that will need to be made in the first pass of the bubble sort.

.....

..... [1]

(c) Bubble sorts make use of two different loops when sorting items into order.

Describe the **two** loops used and their purpose.

1

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

(d) State the name of **one** other sorting algorithm that Poppy could have used.

.....

..... [1]

- (b) Some of Oscar's customers are rated as gold. Customers who are rated as gold are given priority when they make a taxi booking. Some customers rated as gold are shown here.

Arshad	Betty	Dave	Freddie	Harry	Jimmy	Kanwal	Lynn	Siad	Tommy	Will
--------	-------	------	---------	-------	-------	--------	------	------	-------	------

When a customer makes a booking, Oscar will make use of a binary search to check if they are gold rated.

Oscar would like to know if 'Tommy' is gold rated.

- (i) State the **three** values that will be set as the midpoints and then checked against 'Tommy' on each iteration of the binary search.

Show your working here.

Midpoint 1

Midpoint 2

Midpoint 3 [3]

- (ii) Oscar has 75 000 customers stored in his program.

Describe the benefit to Oscar of using binary searches in his program.

Benefit

.....

..... [2]

- (iii) State **one** other search algorithm that Oscar could have used.

..... [1]

- (iv) State the pre-condition which has been met which meant that Oscar did not need to use the search algorithm you stated in question 3(b)(iii).

..... [1]

(iv) A bubble sort has the following complexities:

Best time	$O(n)$
Average and worst time	$O(n^2)$
Worst space	$O(1)$

Describe what each of these complexities mean.

Best time $O(n)$

.....

.....

.....

.....

Average and worst time $O(n^2)$

.....

.....

.....

.....

Worst Space $O(1)$

.....

.....

.....

.....

[6]

(b) (i) Using Big-O notation state the best case complexity of insertion sort.

..... [1]

(ii) Explain what your answer to part (b)(i) means.

.....
.....
.....
.....
.....
.....
..... [3]

(c*) The number of data items in the array is continually increasing.

Insertion sort has a worst case time complexity of $O(n^2)$ and space complexity of $O(1)$.

An alternative sorting algorithm that could be used is bubble sort which also has a worst case time complexity of $O(n^2)$ and space complexity of $O(1)$.

Briefly outline how the bubble sort algorithm works. Discuss the relationship between the complexities and the two sorting algorithms and justify which of the two algorithms is best suited to sorting the array. [9]

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

(c) The target integer 8 exists in a list of integers 1, 4, 6, 9, 8, 12, 15 but is not found during a binary search. There are no errors in the code.

(i) Give the reason why the target integer 8 is **not** found.

.....
..... [1]

(ii) Identify and describe an alternative search algorithm that could be used.

.....
.....
.....
.....
.....
.....
..... [3]

1. A 1-dimensional array stores a set of numbered cards from 0 to 7. An example of this data is shown in Fig in 4.1

2	0	1	7	4	3	5	6
---	---	---	---	---	---	---	---

Fig 4.1

The programmer wants to search for a specific card in the array.

State whether a binary search or a linear search would be the most appropriate method to search for a specific card, and justify your answer.

Search method

Justification

.....

.....

.....

[3]

2. A programmer is developing an ordering system for a fast food restaurant. When a member of staff inputs an order, it is added to a linked list for completion by the chefs.

The user needs to be able to search for, and find, a specific order number.

State an appropriate search algorithm that could be used, and justify your choice against an alternative Search algorithm.

Appropriate Search Algorithm

Justification

.....

.....

.....

.....

.....

[3]

4.

A country's national rail operator provides an app for customers to purchase tickets. An array is used to store the names of the stations on the network. Customers must enter a departure station into the app.

The current contents of the array are shown:

Cavalry	Bridge	Walkway	Museum	Monument	Council House	Theatre	Cinema
---------	--------	---------	--------	----------	---------------	---------	--------

A linear search is used to check if the entered departure station exists in the array.

(i) Identify **one** precondition that is needed before a binary search could be used with the station array.

----- [1]

(ii) A user enters the departure station 'Bridge Heights'

Explain how a linear search would check if the departure station exists in the array.

----- [4]

7. Show how a binary search would be performed on the array shown in Fig. 4.2 to find the value 'duck'.

wolf	monkey	lion	iguana	goat	giraffe	frog	elephant	duck
------	--------	------	--------	------	---------	------	----------	------

Fig. 4.2

[3]

**If you found this
useful, drop a follow
to help me out!**

THANK YOU!

GCST