

4.4 Classification of algorithms Mark Scheme

Q1.

All marks AO1 (knowledge)

Determining if a program will halt;

Max 1 for the following points, but only award mark if 1st mark was awarded:

without running the program; for a particular input;

Q2.

Mark is for AO1 (understanding)

The Halting problem is non-computable / undecideable // there is no algorithm that solves the Halting problem; **A.** it is not computable

In general, inspection alone cannot always determine whether any given algorithm will halt for its given inputs // a program cannot be written that can determine whether any given algorithm will halt for its given inputs;

Max 1 mark



Q3.

All marks AO1 (understanding)

E	Level	Description APERS PR	Mark Range	TICE
	4	A line of reasoning has been followed to produce a coherent, relevant, substantiated and logically structured response. The response covers all three areas indicated in the guidance below and in at least two of these areas there is sufficient detail to show that the student has a good level of understanding. To reach the top of this mark range, a good level of understanding must be shown of all three areas.	10-12	
	3	A line of reasoning has been followed to produce a coherent, relevant, substantiated and logically structured response which shows a good level of understanding of at least two areas indicated in the guidance below.	7-9	
	2	A limited attempt has been made to follow a line of reasoning and the response has a mostly logical structure. At least four points have been made. Either a good level of understanding of	4-6	

[1]

[2]

	one area from the guidance has been shown or a limited understanding of two areas.	
1	A few relevant points have been made but there is no evidence that a line of reasoning has been followed. The points may only relate to one or two of the areas from the guidance or may be made in a superficial way with little substantiation.	1-3

Guidance – Indicative Response

For each guidance point, if the student expands on the point to explain in what way the measure will improve performance then this can be considered to be a second point. For example:

- "Using a processor with more cores" is one point.
- "Using a processor with more cores which will be able to execute multiple instructions simultaneously" is two points.

Note that just "faster" is not enough to count as an expansion point without an explanation of why.

1. Server Hardware

Replace the processor with one which has more cores

Replace the processor with one which has more cache memory // increase the amount of cache memory

Replace the processor with one which runs at a faster clock speed **NE.** faster processor

Use a parallel processor architecture // use more processors which can work in parallel

Use a processor with a bigger word size

Use a processor that makes (better) use of pipelining

Install more RAM // main memory // primary memory

Use RAM // main memory // primary memory with a faster access time

Replace HDDs with SSDs // Replace HDDS with HDDs that can read data at a faster rate

Defragment the HDD

Replace the motherboard with one which has buses which run at a faster clock speed

Replace the motherboard with one which has more lines in the data bus

Use the Harvard architecture

Distribute the processing across multiple servers

2. Network

Replace the network cable with cable that has a higher bandwidth // replace copper cable with fibre-optic cable **A.** Ethernet cable for fibre-optic NE. higher bandwidth network

Replace any wireless / WiFi connections with wired ones

Replace the network cards with ones that can transmit data at a higher bitrate

Consider the overall network design eg how the network is divided into subnets A.

split the network into subnets

Use a star topology (instead of a bus)

Consider using a more efficient protocol for the data across the network

Add additional wireless access points

3. Database and Software

Use a more efficient technique for controlling concurrent access to the database // replace record/table locks with serialisation/timestamp ordering/commitment ordering

Replace the database software with software that uses more efficient algorithms for tasks **A.** examples eg replace linear search with binary search

Use the index feature of the database to speed up searching on fields that are commonly used for this purpose

Rewrite the database software in a language that is suitable for concurrent execution // use a functional programming language for the database software

Ensure the software is compiled rather than executed by an interpreter // rewrite the software in assembly language/machine code

Review the conceptual model of the database to see if it contains any inefficiencies such as data redundancy that could be eliminated **A**. normalise the database design

Consider if it would be appropriate to sacrifice normalisation of the conceptual model to improve performance

Use a non-relational database system A. examples eg NoSQL

Distribute the data across multiple servers

Try to reduce the amount of other (unrelated) software that might be running on the database server at the same time

Try to reduce the number of database accesses that need to be made simultaneously // run some tasks at quiet times / overnight

Purge / archive data that is no longer necessary / in use

EXAM PAPERS PRACTICE

(a) Mark is for AO1 (knowledge)

Merge sort;

(b) Mark is for AO1 (understanding)

4;

Q5.

(a) Mark is for AO1 (knowledge)

 n^2 // O(n^2); **A**. other ways of indicating n^2 e.g. n^2 **A**. On²

1

1

1

[2]

[12]

(b) Marks are for AO1 (understanding)

In each pass through the list n items will be examined; There will be (at most) n passes through the list;

Q6.

Most efficient: B // O(n) C // O(n³) Least efficient: A // O(2ⁿ)

Q7.

The problem can be solved;

In polynomial time or better // in a reasonable amount of time;

- A. "Algorithm exists" for can be solved
- A. Answers relating to space rather than time

Q8.					_
	Proble	m			Tractable? (Tick one)
	The travelling salesm	nan	problem		
	The problem of searching a list.				√ ;
	The Halting problem.				
			_		

A. Alternative indicators for ticks Do not award mark if more than one box is ticked.

Q9.

- (a) Mark is for AO1 (understanding) False;
- (b) Mark is for AO1 (understanding)

THEN Failed \leftarrow True;

(c) All marks for AO1 (understanding) L ← M - 1; Mark as follows: 1 mark: L; 1 mark: ← M - 1; Maximum 1 mark: If not correct 2

[3]

[1]

[2]

1

1

2

- (d) Mark is for AO1 (understanding) $O(k^n);$ **A** kⁿ
- Mark is for AO1 (knowledge) (e) O(log n); A log n
- (f) Mark is for AO1 (knowledge) 0(1);**A** 1
- (g) Mark is for AO1 (knowledge) O(n); **A**n
- (h) All marks AO1 (understanding)

1 mark: As the size of the list increases the time taken to search for an item increases: at the same rate: //

1 mark: A linear search looks at each item in the list in turn (until it reaches the end of the list or the item being searched for is found); so if there are n items in the list the worst case would be n comparisons;



Q10.

All marks AO1 (knowledge) (a)

> **1 mark:** Serial sends one bit at a time / after each other whereas parallel sends multiple bits simultaneously / at same time/

> > _ r



1 mark: Serial uses a single wire / cable / path / line whereas parallel uses several / multiple wires / cables / paths / lines; A serial requires fewer wires

5.

R answers that refer to multiple channels achieved by sharing bandwidth R unless both sides of a point are made.

2

1

1

1

1

2

[10]

(b) Mark is for AO1 (understanding)

Parallel communication can only be used over short distances / / distance between computer and peripheral too great to use parallel communication / / data skew might occur if parallel communication used; To avoid problems of cross-talk / / interference between individual wires; Hardware (for serial communication) is cheaper to manufacture; A fast transmission rate may not be required; Max 1

1

(c) Mark is for AO1 (knowledge)

Number of signal changes per second / / rate at which signals can change; A

Each signal level / signal change represents more than one bit (of data) / / channel supports more than two different signal levels / voltages / / use of modulation / coding technique eg phase modulation; **N.E.** Send more than one bit at a time **Must be clear that there are more than two signal levels**;

Q11.

(a) The problem can be solved // algorithm exists for problem; in polynomial time (or less) // in a reasonable amount of time;

2

2

1

1

1

(b) Use of heuristic;

An algorithm that makes a guess / estimate based on experience; **NE** just algorithm that uses previous knowledge / experience That provides a close-to-optimal solution / approximation // that only works in some cases;

A non-optimal

Example of heuristic method e.g. hill-climbing / stochastic / local improvement / greedy algorithms / simulated annealing / trial and error / any reasonable example;

Relax some of the constraints on the solution; A solve simpler version of problem A limit size of input

MAX 2



R Responses in which more than one row is ticked

A Responses in which a symbol other than a tick is used, so long as it is only placed on one row.

A Use of two symbols, with one indicating which problem is unsolvable and the other indicating which two are solvable, so long as the meaning of the symbols is clear.

(d)

Order of Time Complexity Least Efficient? (Tick one row)

O(2 ⁿ)	√;
O(n)	
O(n ²)	

R Responses in which more than one row is ticked

A Responses in which a symbol other than a tick is used, so long as it is only placed on one row.

A Use of two symbols, with one indicating which algorithm is least efficient and the other indicating which two are more efficient, so long as the meaning of the symbols is clear.

[6]

1

1

Q12.

(a)

Algorithm Name	Requires Sorted List? (Tick one box)
Binary search	✓
Linear search	

1 mark for having a tick in the "Binary search" row.
A alternative indicators for tick eg "Yes"
A a tick for "Binary search" and a cross for "Linear search"
R answers where two ticks have been used.

		•
1	h	1
U	υ	

EY	ΔΜ	DA	DED			Li	st		
	List Length	Outer	Current	Inner	[1]	[2]	[3]	[4]	
		FOINCEL	value	FOINCEL	9	8	5	6	
	4	2	8	1		9			
				0	8				
		3	5	2			9		
				1		8			
				0	5				
		4	6	3				9	
				2			8		
				1		6			

Award **1 mark** for each of the highlighted rectangles which has the correct values written in it in the unshaded cells. Accept responses in which correct values are unnecessarily written out again. Do not award a mark for any rectangle which has an incorrect value written in it.

3

1

1

1

1

(c) The value being moved / CurrentValue / 6 does not need to be put at the start of the list / / should be inserted at position 2 not position 1;
 Because the second condition (in the While statement) is not satisfied;
 MAX 1

(d)

Order of Time Complexity	Tick one box
O(n)	
O(n ²)	¥
O(2 ⁿ)	

A alternative indicators instead of a tick eg a cross, Y, Yes R responses in which more than one box is ticked

- (e) Insertion sort; A Insert sort
- (f) (i) 9, 6, 8; Must be in the order above. Can be separated by any character or a space
- (ii) 9, 20,10; Must be in the order above. Can be separated by any character or a space.

(g)



1 mark for inserting number 4 in the correct place
1 mark for inserting both numbers 3 and 5 in the correct place relative to 4
MAX 1 if any numbers added in the wrong place / any extra numbers added

2

1

2

[11]

Q13.

(b)

(a) Most efficient: C // O(n) A n B // O (n²) A n² Least efficient: A // O(aⁿ) A aⁿ
(i) The problem can be solved; But not in polynomial time // only in exponential (or worse) time // it takes an unreasonable amount of time to do so //

it takes an unreasonable amount of time to do so //
can't be solved quickly enough for it to be useful;
A takes too long for a computer to solve but NE just takes a long time
A "algorithm exists" for can be solved
A answers relating to space rather than time
TO of the solving mark, if states that can be solved in polynomial /
reasonable time

(ii)

Problem	Intractable? Tick One)
The travelling salesman problem.	~
The problem of sorting a list of names into alphabetic order.	
The Halting problem.	

Do not award mark if more than one box is ticked.

Q14.

- (a) Space / Memory (complexity); A amount of memory used
- (b) (i)

N	Pos1	W1	Pos2	W2	Output	
3	1	Rope	1	Rope		
			2	Dagger		1 mark
			3	Rope	Duplicate: Rope	
	2	Dagger	1	Rope		h
			2	Dagger		1 mark
			3	Rope		ſ
	3	Rope	1	Rope	Duplicate: Rope	h
			2	Dagger		
			3	Rope		

A answers which have correct values repeated in empty cells, but do not award a mark if there are any incorrect values within the block for which the mark is being awarded.

A additional rows in trace table, so long as the trace is correct. **DPT** if just "Duplicate" or "Rope" are written in the Output column when it should be "Duplicate: Rope" or if the value of Pos1 is written in the output instead of W1 e.g. 1 instead of "Rope"

If candidate has not written in the value of N, only one mark should be lost (for the top rectangular area) for this mistake

(ii) O(n²);

1

3

(iii) Marks can only be awarded if correct answer for part (b)(ii) Alternative 1:

Algorithm has nested loops // two loops with one inside the other; **A** reference to inner and outer loops Each loop repeats N times;

Alternative 2:

The (basic) operation / If statement / file read / comparison is carried out N^2 times; because it is inside nested loops // because each loop executes N times;

Alternative 3:

Each of the (N) entries is compared to each of the (N) others // each entry is compared N times; so N2 (A N*N) comparisons/operations are

1

[4]

required // N*N=N²; **A** uppercase or lowercase n **A** answers where examples are used instead of N and N², e.g. 3 and 9. **A** check as alternative to comparison

Max 2

2

[7]

Q15.

- (a) Is it possible in general to *write a program / algorithm;* that can tell, given any program and its inputs and without *running / executing the program;*, whether the given program with its given inputs will halt?
 A "it" in second reference to program.
 - A "create a Turing machine" for "write an algorithm"
- (b) Shows that some problems are non-computable / undecideable // shows that some problems cannot be solved by a computer / algorithm;

In general, inspection alone cannot always determine whether any given algorithm will halt for its given inputs // a program cannot be written that can determine whether any given algorithm will halt for its given inputs; A it is not computable



Max 1

3

1

[3]



Q16.

(a)				
. ,	Position	Value	Order Examined In	
	8	Philip	1	;
	10	Ravi	3	
EX/	11	Richard	4	
	12	Timothy	2	/

mark for row 8 correct
 mark for row 11 correct
 mark for both rows 10 and 12 correct
 Do not award mark for a particular number if same number is written more than once

(b)

8

(c) Order of complexity	Tick one box
O(log₂ n)	~
O(n)	

O(n²)	
-------	--

Do not award mark if more than one box ticked

[5]

1

Q17.

(a) (i) $O(a^n)$; **A** exponential, a^n 1 (ii) A; 1 (b) (i) The problem can be solved // algorithm exists for problem; But it cannot be solved in polynomial time // but not quickly enough to be useful: It takes an unreasonable amount of time; to solve; A too long time but R long time 2 (ii) Use of heuristic; algorithm that makes a guess based on experience; That provides a close-to-optimal solution/approximation; that only works in some cases; A non-optimal Example of heuristic method e.g. hill-climbing / stochastic / local improvement / greedy algorithms / simulated annealing / trial and error / any reasonable example; Relax some of the constraints on the solution; A solve simpler version of problem Max 2 Q18 A procedure that is defined in terms of itself;

A A procedure that calls itself R re-entrant

(b) Store return addresses;
 Store parameters;
 Store local variables/ return values;

Max 1

1

[6]

(C)

Number	Entry	Output
11	1	
11	2;	
11	3;	
11	4;	4;

		4
(d)	A linear search// To find/output the position/index of Number in Items;	1
(e)	Number is not an entry in Items// Stack overflows;	1
(f)	Test for reaching the end of Items;	1
(g)	Binary Search; An iterative solution;	Max 1

[10]



Examiner reports

Q1.

The Halting Problem and Turing Machines were the focus of this question. A number of answers suggested that students had seen a simulation of a Turing Machine but hadn't fully understood what they had seen.

Q2.

The Halting Problem and Turing Machines were the focus of this question. A number of answers suggested that students had seen a simulation of a Turing Machine but hadn't fully understood what they had seen. A number of answers talked about the Halting problem being intractable rather than non-computable.

Q3.

A very good range of responses was received to this question, with approximately half of students achieving five or more marks. Most students addressed all three aspects of the question (hardware, network, database and software). Students tended to make more points about how the hardware could be improved than about the other two areas. This was acceptable but students needed to have covered all three areas to achieve a mark of ten or above.

Some students wrote too vaguely to achieve marks, for example by writing that a "faster processor" would improve performance, without referencing a factor such as the clock speed that would make the processor faster. Other mistakes included believing that the question required students to contrast thin-client and thick-client and that the system was web based.

A small number of students wrote about issues which might be causing the system to perform poorly instead of explaining how the performance of the system could be improved. Such responses were not worthy of a mark.

XAM PAPERS PRACTICE

This was the Section A question that students found hardest, with very few getting full marks. Not many students were able to identify the time complexity of either merge sort or (to a lesser extent) bubble sort and a significant number of students thought that the binary search and/or the post-order tree traversal would not be used to solve tractable problems.

When students could state the time complexity of the bubble sort algorithm they were rarely able to clearly explain why $O(n^2)$ was the correct answer.

Q5.

This was the Section A question that students found hardest, with very few getting full marks. Not many students were able to identify the time complexity of either merge sort or (to a lesser extent) bubble sort and a significant number of students thought that the binary search and/or the post-order tree traversal would not be used to solve tractable problems.

When students could state the time complexity of the bubble sort algorithm they were rarely able to clearly explain why $O(n^2)$ was the correct answer.

Q11.

This question was about solvability, tractability and efficiency.

Part (a) was well answered, with around two thirds of students achieving both marks for explaining what a tractable problem was. Some students achieved the second mark by stating that the problem could be solved in a reasonable timeframe, and it would have been nice to see a more technical explanation of what this meant, ie that the problem could be solved with polynomial time efficiency or better.

Just over half of students achieved at least one more mark for explaining how an intractable problem might be tackled in Part (b). A common misconception was that brute force could be used to try out every possible solution, which would clearly not achieve a solution in a reasonable timeframe. Some responses referred to making a guess at an answer, but did not explain that some experience or knowledge would need to be applied to make this an educated guess, so missed out on the associated mark.

Parts (c) and (d) were both well answered, with over three quarters of students achieving the mark for each question part.

Q12.

- (a) The overwhelming majority of students were able to correctly identify that it was the binary search algorithm that required the list to be sorted for this part.
- (b) The trace for this part was also well completed with about three quarters of students getting some marks and well over half getting full marks.
- (c) For this part, around half of the candidates were correctly able to explain that the value of InnerPointer did not decrease to zero because either the second while loop condition was not
- (d) For this part, about two thirds of students correctly identified that the algorithm that they had traced was of time complexity $O(n^2)$.

EX

This part was poorly answered, with only about one third of students correctly identifying that the algorithm they had traced was an insertion sort. Bubble sort was a far more common but incorrect response.

- (f) Parts (i), (ii) were all well answered. The most common error in both parts of was to perform a traversal of the tree instead of using it as a binary search tree.
- (g) This part was all well answered. The most common error in both parts of was to perform a traversal of the tree instead of using it as a binary search tree.

Q13.

For (a), over two thirds of the candidates correctly ranked the algorithms by order of complexity.

For (b)(i), the majority of candidates explained that an intractable problem could not be solved in polynomial time or better. However, many failed to make clear that the problem could be solved at all, so only achieved one of the two marks.

Part (b)(ii) was well answered with over two thirds of candidates correctly recognising that the intractable problem was the Travelling Salesman problem.

Q14.

Part (a): This question part was well answered, with most students correctly identifying that space complexity was the second measure of complexity (in addition to time complexity).

Part (b)(i): In this question part, students were required to carry out a trace of a simple algorithm for looking for duplicate values in a file. The primary purpose of this question part was to give students the opportunity to study the algorithm so that they could discuss its complexity in question parts (b)(ii) and b(iii). The majority of students achieved all three marks for fully completing the trace table. However, disappointingly, approximately a third of students scored no marks at all for what was a relatively straightforward trace to complete.

Part (b)(ii): Pleasingly, approximately two thirds of students recognised that the algorithm had order of time complexity $O(n^{2})$

Part (b)(iii): Students justified the order of time complexity as being $O(n^2)$ in various ways. Some did it by analysing the code and recognising that there was a for loop nested inside another for loop, with each loop repeating n times. Others did it by considering the number of comparisons made and recognising that each of the n items in the file would be compared to each other item, resulting in n² comparisons being made. The most commonly made mistake was to refer to there being two loops in the algorithm, without making clear that these were nested. Some responses were too superficial to be creditworthy, failing to relate the complexity to the steps involved in the algorithm.

Q15.

Part (a): Most students got at least one mark for this question part, usually for identifying that the first required response was "write a program". Many also went on to achieve the second mark too for identifying that the program being tested would not be run.

Part (b): This question part was poorly attempted. Students needed either to explain that the Halting problem was an example of a non-computable problem, or to state that there was no solution to the problem, so inspection alone could not always determine if a program would halt on a given set of inputs. Some students stated that the Halting problem could be used to show if a program / algorithm was computable or not. This is not the case, as there is no solution to the Halting problem. Rather, it can be used to demonstrate that some problems are not computable as it is an example of one such problem. A small number of students confused non-computability with intractability.

Q16.

Part (a): The binary search method was well understood and the majority of candidates were able to correctly label the sequence of four items that would be checked to search for the name "Richard". Some candidates missed out on the final mark by not realising that there would be a fourth comparison, i.e. "Richard" compared with "Richard". A minority of candidates applied the linear search method instead, labelling each of the names from Adam down to Richard consecutively from 1 to 11.

Part (b): This part was not tackled as well as part (a). Most candidates seemed to realise that the answer involved logarithms or powers of two, but the most common response was seven rather than eight, the correct answer. When calculating the number of comparisons required to search a list of n items, $\log_2(n)$ should be calculated and the result then rounded up. So $\log_{2(137)=7.010.2 \text{ decimal places, which rounds up to 8.}$

Part (c): This question part was well answered, with over half of the candidates correctly identifying the complexity of the binary search method as $O(\log_{2n})$.

Q17.

Part (a): Most candidates were able to identify both the time complexity of the algorithm from the graph and also to indentify which of the three algorithms was the most time-efficient.

Part (b): An intractable problem is one which has a solution, but this solution cannot be found by any algorithm that has polynomial time complexity or better. Most candidates identified the fact that a solution to an intractable problem could not be found in polynomial time so gained one mark but only a small number went on to explain that the problem could be solved to gain the second mark.

Many candidates scored at least one mark for their description of how an intractable problem might be 'solved', with the most common response being that heuristic knowledge would be applied. Some candidates went on to explain what this meant to get a second mark. Other good points included simplifying the problem itself by relaxing some constraints and using an algorithm that produced an acceptable but non-optimal solution. Answers which simply stated that guesswork would be used did not gain credit, as a heuristic is more than simply guessing an answer.

Q18.

Candidates generally scored well on this question. Recursively-defined was well understood although many candidates were unable to describe the use of the stack well enough. It was pleasing to see the majority of candidates obtaining most of the marks on part (c). Candidates often failed to obtain the mark for part (d) due to inadequate descriptions. Although many candidates provided a situation where the algorithm will fail, fewer were able to suggest a suitable modification. Once again this was often due to an inability to express themselves well. A wide range of answers were supplied for part (g) but a substantial number of correct responses were given.

