**Victorian Certificate of Education
2023**

Letter

STUDENT NUMBER

ALGORITHMICS (HESS)
Written examination

Wednesday 15 November 2023

**Reading time: 3.00 pm to 3.15 pm (15 minutes)
Writing time: 3.15 pm to 5.15 pm (2 hours)**

QUESTION AND ANSWER BOOK

Structure of book

Section	Number of questions	Number of questions to be answered	Number of marks
A	20	20	20
B	10	10	80
			Total 100

- Students are permitted to bring into the examination room: pens, pencils, highlighters, erasers, sharpeners, rulers and one scientific calculator.
- Students are NOT permitted to bring into the examination room: blank sheets of paper and/or correction fluid/tape.

Materials supplied
- Question and answer book of 29 pages
- Answer sheet for multiple-choice questions

Instructions
- Write your **student number** in the space provided above on this page.
- Check that your **name** and **student number** as printed on your answer sheet for multiple-choice questions are correct, **and** sign your name in the space provided to verify this.
- All written responses must be in English.

At the end of the examination
- Place the answer sheet for multiple-choice questions inside the front cover of this book.

Students are NOT permitted to bring mobile phones and/or any other unauthorised electronic devices into the examination room.

SECTION A – Multiple-choice questions

<div style="border:1px solid black">

Instructions for Section A

Answer **all** questions in pencil on the answer sheet provided for multiple-choice questions.

Choose the response that is **correct** or that **best answers** the question.

A correct answer scores 1; an incorrect answer scores 0.

Marks will **not** be deducted for incorrect answers.

No marks will be given if more than one answer is completed for any question.

Use the Master Theorem to solve recurrence relations of the form shown below.

$$T(n) = \begin{cases} aT\left(\dfrac{n}{b}\right) + kn^c & \text{if } n > 1 \\ d & \text{if } n = 1 \end{cases} \quad \text{where } a > 0,\, b > 1,\, c \geq 0,\, d \geq 0,\, k > 0$$

$$\text{and its solution } T(n) = \begin{cases} O(n^c) & \text{if } a < b^c \\ O(n^c \log n) & \text{if } a = b^c \\ O(n^{\log_b a}) & \text{if } a > b^c \end{cases}$$
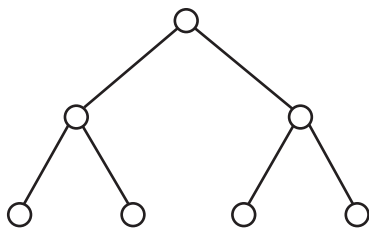
</div>

Question 1

Which one of the following is the signature specification for the 'push' operation of a stack?

A. stack → stack

B. stack → element

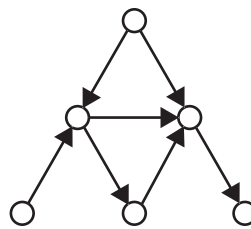C. element × stack → stack

D. element × stack → element

Question 2

Which one of the following graphs is a directed acyclic graph?
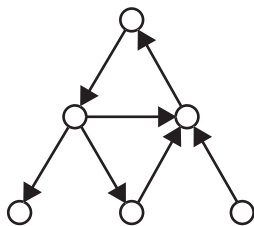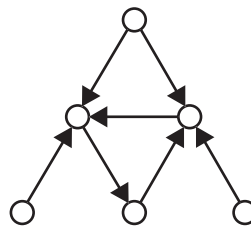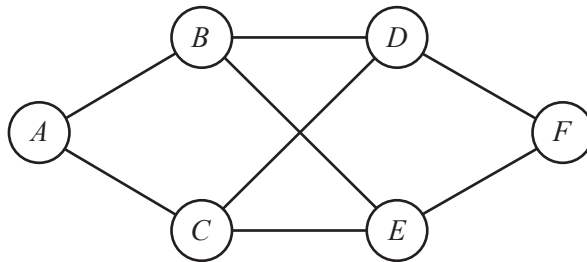
A.

B.

C.

D.

DO NOT WRITE IN THIS AREA

Question 3

What do the non-leaf nodes in a decision tree represent?

A.   the different answer options to decision questions

B.   the results of the problem

C.   the decision questions

D.   the distance from the root node

Question 4

Consider the following graph.

Depth-first search is run on the graph, starting at node A.

The order in which the nodes are traversed could be:

A. A, B, C, D, E, F

B. A, B, C, D, F, E

C. A, B, D, C, E, F

D. A, B, D, C, F, E

Question 5

Which one of the following graph algorithms repeatedly selects the lowest-cost edge that connects out from a connected component of nodes?

A.   the Bellman-Ford algorithm

B.   Dijkstra's algorithm

C.   the Floyd-Warshall algorithm

D.   Prim's algorithm

2023 ALGORITHMICS EXAM
Question 6

Consider the following algorithm.

```
Algorithm graph_sum(G, u, total):
  Mark u as visited
  If there are no unvisited nodes adjacent to u Do
    Return total
  Select v as an unvisited node adjacent to u connected by
      the most expensive edge
  Return graph_sum(G, v, total + cost(u,v))
```

What is the value returned by the *graph_sum* algorithm when executed on the graph, *G*, shown below with *u = A* and *total = 0*?

A. 16
B. 21
C. 22
D. 26

SECTION A – continued
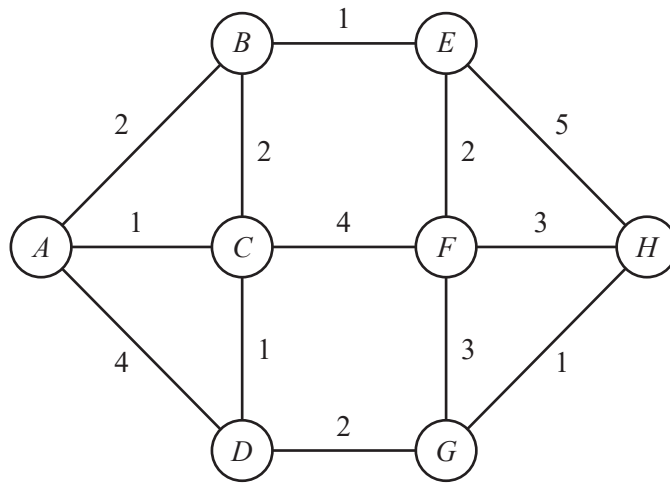
Question 7

The Floyd-Warshall algorithm is being used to find the transitive closure of the following graph.

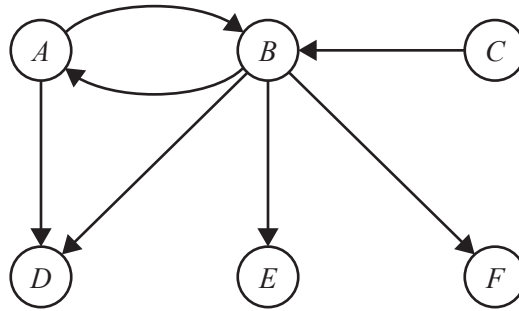The outer loop of the algorithm has already iterated over node A.

Which of the following edges is **not** added when next iterating over node B?

A. (A, C)

B. (A, E)

C. (C, A)

D. (C, D)

Question 8

Sam is creating a maze-solving program that takes a satellite photo of a maze as input and returns the path from the entrance to the centre as output. Image recognition will be used to identify features of the maze, such as its walls, entrance and centre.

Which algorithm is likely to be most efficient for Sam to use to find the path from the maze's entrance to its centre?

A. breadth-first search

B. depth-first search

C. Dijkstra's algorithm

D. the A* algorithm

Question 9

The following naive algorithm calculates the n'th term in the Fibonacci sequence using recursion to calculate the $(n-1)$'th and $(n-2)$'th terms. For example, 13 would be the 8th term in the sequence 0 1 1 2 3 5 8 13…

```
Algorithm fib(n):
  If n = 1 Do
    Return 0
  If n = 2 Do
    Return 1
  Return fib(n-1) + fib(n-2)
```

How many redundant recursive calls would be made when *fib* is called with *n = 6*?

A. 5

B. 9

C. 14

D. 18

Question 10

A dynamic programming algorithm is used to solve a change-making problem with the coin denominations of 1, 3 and 4. This algorithm iteratively solves the sub-problem of finding the fewest coins required to give k change. The algorithm is used to solve the problem for a total amount of 6.

What are the final values stored in the array used by the algorithm?

A. [0, 1, 2, 1, 1, 2, 2]

B. [0, 1, 1, 3, 4, 4, 3]

C. [0, 1, 1, 3, 4, 1, 3]

D. [3, 3]

Question 11

While it is working, a robot vacuum cleaner tracks its location relative to its charging dock. When it is running low on power, it navigates back to the dock.

Which one of the following algorithmic approaches for navigating back to the charging dock best describes the approach of hill climbing on a heuristic function?

A. The robot navigates to the charging dock by retracing the path it took from the charging dock to its present position.

B. The robot navigates to the charging dock by calculating an optimal path back to the charging station using its knowledge of locations that it previously traversed.

C. The robot calculates the distance to the charging dock of a randomly selected open position adjacent to its current position. It moves to that position only if the position is closer to the charging dock. It repeats this process until the robot reaches the charging dock.

D. The robot calculates the distance to the charging dock of a randomly selected open position adjacent to its current position. If that position is closer to the charging dock, it always moves there. If that position is further away from the charging dock, it sometimes moves there, based on some probability function that slowly decreases the likelihood of such moves over time. It repeats this process until the robot reaches the charging dock.

Question 12

Which one of the following families of functions has the largest Big-O complexity?

A. $O(100n^2 + 1000000n)$

B. $O(2n^{10})$

C. $O\left(5000\log_2(n)\right)$

D. $O\left(2000 \cdot (0.5)^n\right)$

Question 13

The recurrence relationship $T(n)$ gives the running time of an algorithm with respect to the size of its input, n, where

$$T(n) = \begin{cases} 9T\left(\dfrac{n}{2}\right) + 4n^3 & n \geq 1 \\ 500 & n = 0 \end{cases}$$

What is the Big-O time complexity of the algorithm with respect to n?

A. $O(n^3)$

B. $O\left(n^3 \log(n)\right)$

C. $O\left(n^3 \log_2(9)\right)$

D. $O\left(n^{\log_2(9)}\right)$

Question 14

The worst-case time complexity when executing quicksort on a list that is already sorted in descending order will arise

A. only when the first item is chosen as the pivot each time.
B. only when the last item is chosen as the pivot each time.
C. when the first or the last item is chosen as the pivot each time.
D. when the pivot is chosen at random.

Question 15

A small multilayer perceptron neural network is shown below.
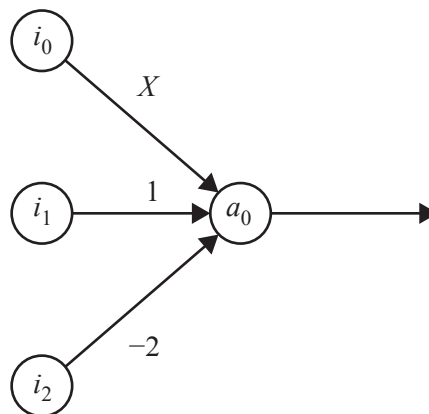
When the input nodes i_0, i_1 and i_2 are set to 0.5, 0.5 and 1, respectively, what value of X makes the activation of node a_0 equal to 0?

A. -2
B. 0
C. 1
D. 3

Question 16

Consider the data points below, each of which belongs to one of two categories, ×'s and ○'s.

Which one of the lines best represents the support vector machine classifier for this data?

A. A

B. B

C. C

D. D

Question 17

Which one of the following statements is **incorrect**?

A. Undecidable problems are decision problems.
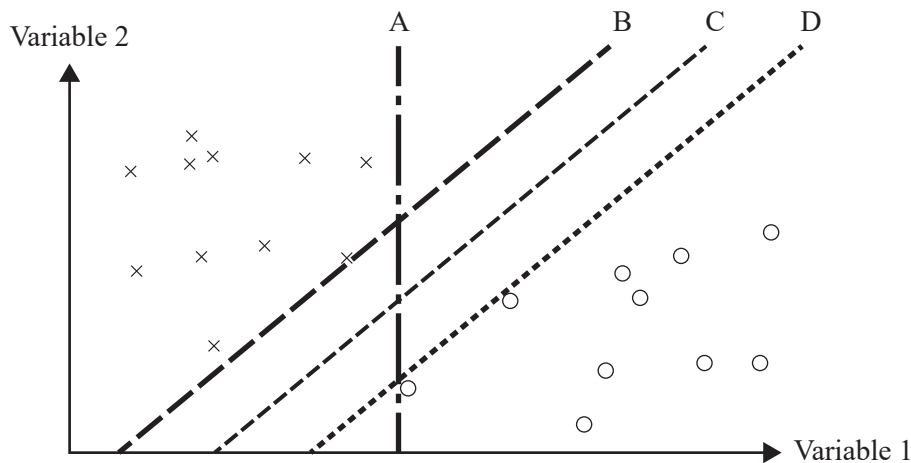
B. No instance of an undecidable problem can ever be solved.

C. There exist algorithms that can solve many instances of an undecidable problem.

D. There is no algorithm that correctly solves every instance of an undecidable problem.

Question 18

Which one of the following is a correct description of problems in the P-time complexity class?

A. They can be solved deterministically in polynomial time with respect to the size of the input.

B. They cannot be verified deterministically in polynomial time with respect to the size of the input.

C. They can be solved deterministically in polynomial time with respect to the size of the output.

D. They can be verified deterministically in polynomial time with respect to the size of the output.

Question 19

Which one of the following statements about Turing machines is correct?

A. It is possible for a specific Turing machine to halt on all inputs.

B. No specific Turing machine halts on all inputs.

C. All Turing machines do not halt on all inputs.

D. All Turing machines halt on some inputs.

Question 20

Which one of the following problems is **not** in the NP-Hard complexity class?

A. finding a solution to the 0-1 knapsack problem

B. finding the minimum cost spanning tree of a graph

C. finding a colouring of a graph with at most four colours

D. finding the minimum cost solution to the travelling salesman problem

END OF SECTION A
TURN OVER

SECTION B

<div style="border:1px solid">

Instructions for Section B

Answer **all** questions in the spaces provided.

Use the Master Theorem to solve recurrence relations of the form shown below.

$$T(n) = \begin{cases} aT\left(\dfrac{n}{b}\right) + kn^c & \text{if } n > 1 \\ d & \text{if } n = 1 \end{cases} \qquad \text{where } a > 0,\, b > 1,\, c \geq 0,\, d \geq 0,\, k > 0$$

$$\text{and its solution } T(n) = \begin{cases} O(n^c) & \text{if } a < b^c \\ O(n^c \log n) & \text{if } a = b^c \\ O(n^{\log_b a}) & \text{if } a > b^c \end{cases}$$

</div>

Question 1 (2 marks)

Describe **one** motivation for using functional abstractions when designing algorithms.

_____

_____

_____

_____

_____

Question 2 (8 marks)

Amir is developing a program to assist with planning projects. Projects are composed of tasks. Tasks take a number of minutes to complete and require a certain number of people. They may also have one or more prerequisite tasks, which are other tasks that must be completed beforehand.

a. Describe a data structure that could be used to model a project. Explain how the features of the project would map to the data structure. 3 marks

_____

_____

_____

_____

_____

_____

b.   Write the signature specification for adding a new task to a project.   2 marks

_____

_____

c.   The *isPrereq*(*P*, *a*, *b*) function returns *True* if, in project *P*, task *b* is an immediate prerequisite of task *a*.

Write pseudocode for an algorithm *requiresTask*(*P*, *a*, *b*) that returns *True* if task *b* must be completed before task *a*, and *False* otherwise. It takes the following arguments:

- a project, *P*
- two tasks, *a* and *b*

Your algorithm should make use of the *isPrereq* function.   3 marks

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Question 3 (5 marks)

A commercial property company owns several shopping centres. A typical shopping centre has between 200 and 400 shops, and some have over 500 shops. A facial recognition system that tracks which shops people visit has recently been installed in all of the shopping centres. Every time a person visits a shopping centre, the tracking algorithm outputs a trip for that visitor as a sequence of shops that were visited.

a. Alexie initially considers storing a tally of the trip data in a dictionary with an entry in the dictionary for every possible trip to each shopping centre.

   Outline **one** issue with this data model that would prevent it from being used. 1 mark

   _____

   _____

   _____

b. Alexie proposes to aggregate the shopping trip data by creating a directed graph for each shopping centre. A node is added to the graph for each shop and a special node is added to represent 'outside' the shopping centre. A directed edge is created between each ordered pair of nodes. The edge (u, v) is given a weight equal to the number of trips in which a visitor visits v directly after u.

   i. Describe how the most-visited shop or shops could be determined from this data model. State the Big-O time complexity of this operation with respect to n, the number of shops in a particular shopping centre. 2 marks

   _____

   _____

   _____

   _____

   _____

   ii. This data model is criticised because it is no longer possible to determine how many visitors visit a particular pair of shops anywhere in their trip.

   Explain how this data model could be modified so that the company can determine how many visitors visit any particular pair of shops. 2 marks

   _____

   _____

   _____

   _____

Question 4 (9 marks)

a. Define the class of complete graphs and draw an example of a complete graph with five nodes.   2 marks

_____

_____

_____

_____

A graph that is formed from a subset of nodes of a graph and all the edges from the original graph that connect pairs of those nodes is called an induced subgraph.

b. The nodes of a complete graph are divided into two sets, *U* and *V*.

Explain why the two subgraphs induced by these two sets of nodes are also complete graphs.   2 marks

_____

_____

_____

_____

c.   **i.**   Consider the class of graphs created by taking an undirected complete graph with k nodes and then giving each edge in the graph a direction to create a directed graph. Let G be one such graph.

The following algorithm finds a path containing every node in G.

```
Algorithm findPath(G):
1    If G is empty Do
2      Return an empty list
3    If G has one node, v Do
4      Return a list containing v
5    Select a node v from G at random
6    Create a set V_in containing all nodes u for which
       (u,v) exists
7    Use the nodes V_in to create an induced subgraph
       called G_in
8    Create a set V_out containing all nodes u for which
       (v,u) exists
9    Use the nodes V_out to create an induced subgraph
       called G_out
10   in_path ← findPath(G_in)
11   out_path ← findPath(G_out)
12   path ← create a new list by joining together in_path,
       v and out_path in sequence
13   Return path
```

The *findPath* algorithm is executed on the graph shown below.
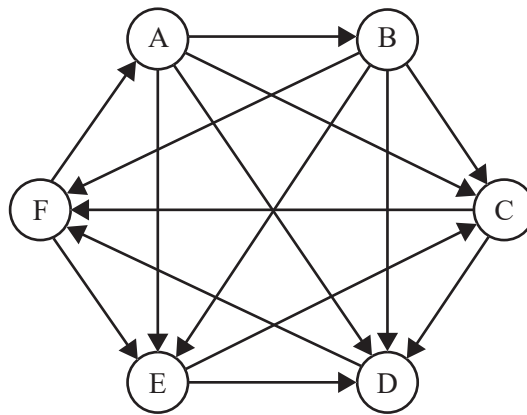
The node A is randomly selected in line 5 of the algorithm.

Find the values of *V_in* and *V_out* after the algorithm has executed up to the start of line 10.   1 mark

ii. Show by induction that every graph in the class of directed graphs defined in **part c. i.** contains a path containing every node. 4 marks

___

___

___

___

___

___

___

___

___

___

___

___

___

___

Question 5 (7 marks)

The PageRank algorithm has been found to accurately rank urban spaces in terms of their human traffic density. Urban spaces can be modelled using graphs, with nodes representing urban spaces such as parks, squares, shops and playgrounds, and edges representing the connectivity between them. David decides to model the area around his local shops using the graph shown below.
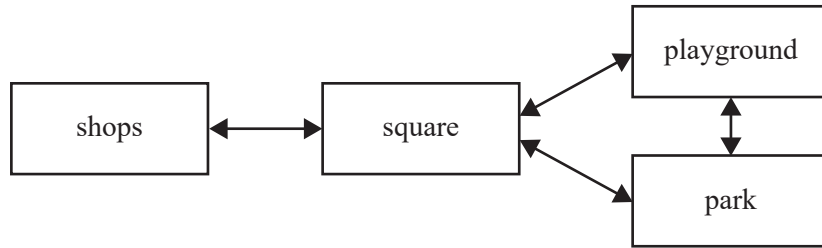
a. Explain how the problem of estimating the importance of urban spaces based on their connectivity can be analogous to the problem of ranking the importance of web pages. 2 marks

_____

_____

_____

_____

_____

_____

b. What PageRank value would each node in David's graph be initialised to? 1 mark

_____

c.   Let *nodePR*(*G, r, u*) be a function with the following arguments:
- *G*, a network graph of urban spaces
- *r*, the PageRanks calculated in the previous iteration of the PageRank algorithm for each node in *G*, given as a dictionary
- *u*, a node in *G*

nodePR(*G, r, u*) returns the updated PageRank of node *u*.

Write pseudocode for an algorithm *pageranks*(*G*) that takes one input, *G*, a network graph of urban spaces. It returns a dictionary containing a (*node, PageRank*) pair for each node in the graph *G*. Your algorithm can assume that there are no nodes with zero outgoing edges, as urban spaces always have a way to leave.   4 marks

Question 6 (10 marks)

A building services engineer is designing the plumbing for a new apartment building. The building will have a central hot-water system serving all the apartments. This requires a network of pipes that form a spanning tree that connects every apartment to the central hot-water system.

Hot-water pipes are expensive, and so the overall length of pipe to be used in the design needs to be minimised. The distance from the central hot-water system to each apartment also needs to be minimised so that less water is wasted each time a resident needs hot water.

Consider a graph that has the central hot-water system and each apartment as nodes, and all possible pipe routes as edges, with edge weights indicating the length of each pipe.

a. Explain why Prim's algorithm would be appropriate for designing an optimal hot-water network of pipes if only the total cost of the pipes needed to be considered. 1 mark

b. Let *Prim*(*G*) be a function that takes as input a weighted graph, *G*, and returns a minimum spanning tree.

Write pseudocode for an algorithm that returns all the possible minimum spanning trees of any given weighted graph. 4 marks

c. Instead, if the only goal was to minimise the length of pipe from the central hot-water system to each apartment, what algorithm would be most suitable for designing the pipe network? Explain why your chosen algorithm best meets the requirements of this problem.   2 marks

_____

_____

_____

_____

_____

d. Describe an algorithmic approach to finding which pipes to use for connecting the apartments that appropriately considers both of the desired goals: minimising the total amount of piping and minimising the length of pipe from the central hot-water system to each apartment.   3 marks

_____

_____

_____

_____

_____

_____

_____

Question 7 (8 marks)

A machine is designed to organise balls of distinct sizes from smallest to largest. Given a bag containing an unknown number of balls, possibly zero, the machine follows these steps:

1. … <a missing step here> …
2. Randomly draw two balls from the bag. Let the smaller ball be A, having diameter d_A, and the larger ball be B, having diameter d_B.
3. Divide all remaining balls in the bag into three new bags, named *Bag1*, *Bag2* and *Bag3*, using the following criteria:
    a. All balls whose diameter is less than d_A are put into *Bag1*.
    b. All balls whose diameter is between d_A and d_B are put into *Bag2*.
    c. All balls whose diameter is greater than d_B are put into *Bag3*.
4. Recursively organise the balls in *Bag1*, *Bag2* and *Bag3*.
5. Output the balls in the following sequence: the sequence of balls in *Bag1*, *A*, the sequence of balls in *Bag2*, *B,* the sequence of balls in *Bag3*.

a. Describe the base case of the recursive ball-organising process that is missing in Step 1.   2 marks

The machine has a device to read the diameter of a ball in constant time. Once the diameter of a ball is known, the machine will then be able to put it into the appropriate bag.

b. Consider the case where every time the machine is asked to organise balls, in Step 3 it creates three bags with a similar number of balls.

   i. Let $S(n)$ be the time complexity for the machine to organise a bag of n balls in this case.

   Explain why $S(n)$ is given by $S(n) = 3S\left(\dfrac{n}{3}\right) + O(n)$ for $n > 1$.

   1 mark

ii. Solve $S(n)$, giving your solution in Big-O notation. 2 marks

_____

_____

_____

_____

_____

c. Consider the case where every time the machine is asked to organise balls, in Step 3 it puts all the balls into only one bag.

    i. Let $T(n)$ be the time complexity for the machine to organise a bag of n balls in this case.

        Explain why $T(n)$ is given by $T(n) = T(n-2) + O(n)$ for $n > 1$. 1 mark

_____

_____

_____

_____

    ii. Deduce the worst-case time complexity of the ball-organising process in this case and state it using Big-O notation. 2 marks

_____

_____

_____

_____

_____

Question 8 (7 marks**)**

A local community tracks data on successful tomato production in its area. A key requirement for a successful crop is rainfall, which is measured relative to the average rainfall in the area on a scale from −10 to +10. The graph below shows data for a sample of tomato crops, with relative rainfall shown by the scale, and success or failure of the crop indicated by the symbol used for each sample.
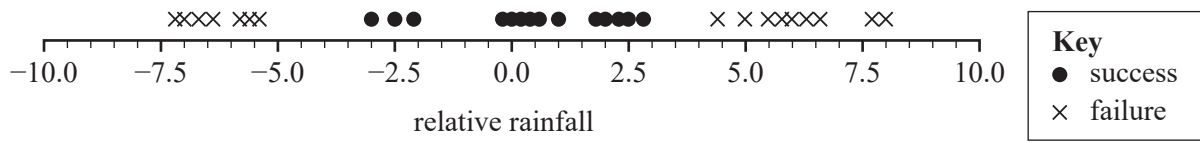
a. Explain how a support vector machine (SVM) could be used as a classifier for this data. 2 marks

_____

_____

_____

_____

_____

_____

_____

_____

The community is keen to analyse the effect of other factors on tomato production, such as soil type and average temperature. At a community meeting there is a proposal to have at least 1000 local residents provide relevant data, which would then be used to train a neural network. The community hopes that this model can then be used to predict and explain successful tomato production, and that they can sell the model to other communities for use in their areas.

b.   Describe one technical and one ethical risk with this approach to developing a tomato production model.   2 marks

Technical risk _____

_____

_____

Ethical risk _____

_____

_____

Years later, the community's idea is hugely successful and results in an online tool, *Perfect Tomatoes,* which can correctly classify the tomato production capability of any region in the world via a real-time natural language interface. Some people in the community claim that this is evidence of artificial intelligence.

c.   Explain whether *Perfect Tomatoes* would be considered artificial intelligence with respect to **two** different conceptions of artificial intelligence.   3 marks

_____

_____

_____

_____

_____

_____

_____

Question 9 (8 marks)

Sanvi shows an old magazine article to her friend, Jade.

Sanvi: 'Hey! Computers can do anything! Look – I found an ancient reference that says so!'

A *Time* magazine article from 1984 quotes a software editor as follows.

> 'Put the right kind of software into a computer, and it will do whatever you want it to. There may be limits on what you can do with the machines themselves, but there are no limits on what you can do with software.'
>
> Source: AL Taylor, 'The Wizard Inside The Machines', *Time*, 16 April 1984, pp. 58 and 59

Sanvi: 'With nearly 40 years of hardware improvements since then, it must be even truer now!'

Jade: 'Not so fast. I think the statement in *Time* magazine is wrong'.

a. Explain why the existence of undecidable problems shows that the *Time* magazine quote is incorrect, and provide a specific example of such a problem.   3 marks

_____

_____

_____

_____

_____

_____

_____

Sanvi is still unsure and seeks further clarification.

Sanvi: 'Wait, I thought undecidable problems were all about the basic foundations of
          mathematics, not about computing.'

b.   With reference to Hilbert and Ackermann's *Entscheidungsproblem*, explain the historical
          connection between the foundations of mathematics and the computing concept of
          undecidability.   3 marks

_____

_____

_____

_____

_____

_____

_____

_____

Sanvi keeps going and challenges her friend's first response.

Sanvi: 'The problems that we know are undecidable seem too theoretical and not at all
          like problems that we actually use computers to solve. So apart from these artificial
          undecidable problems, anything can be solved. Isn't that right?'

Jade: 'Again, not so fast. Sometimes we can, sometimes we can't.'

c.   Explain why Jade is right.   2 marks

_____

_____

_____

_____

_____

_____

SECTION B – continued
TURN OVER

Question 10 (16 marks)

A gym offers group fitness programs, each running on a weekly timetable for a full term. The gym employs a number of instructors to run the programs. The gym has many workout rooms and studio rooms, containing exercise equipment and empty floor space respectively. Gym members may enrol in multiple programs each term. The gym has finalised its program enrolments for the first term and now needs to establish a timetable.

The week is divided up into six timetable blocks, and a set of programs will be scheduled to each block. Within each block, however, an instructor, gym member or room cannot be part of more than one program. The gym requires assistance in allocating programs to a block. The blocks are shown in the weekly timetable below.

	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
Early morning		Block 1	Block 1	Block 5	Block 4	Block 4	
Morning	Block 2	Block 5	Block 2	Block 2	Block 2	Block 5	Block 5
Afternoon	Block 3	Block 6	Block 3	Block 3	Block 3	Block 6	Block 6
Evening		Block 4	Block 4	Block 6	Block 1	Block 1	

Enrolment data for each program is recorded in a table. An example table is provided below; note that the actual table would include many more programs and students.

Program number	Program	Instructor initials	Room requirements	Enrolments (member ID)
1	Beginner Yoga	KEB	studio room	3729 8001 5057
2	Strength Training	SBA	workout room	6756 5057 2214
3	Mixed Martial Arts	ARD	studio room	8771 2214 9090
4	Pilates	KEB	workout room	7862 3333 4089

a.   Draw a network graph to represent the example table, with program numbers as nodes and
         conflicts as edges.   1 mark

b.   Describe the graph colouring problem.   1 mark

_____

_____

_____

c.   Explain how the problem of allocating programs to blocks in the gym's timetable without
         conflicts can be mapped to the graph colouring problem.   2 marks

_____

_____

_____

_____

_____

_____

TURN OVER

d. The gym needs to find an allocation of programs to timetable blocks.

State the complexity class of this problem and explain what features of the problem give rise to this classification. 2 marks

e. As an advisor to the gym, consider the backtracking and simulated annealing approaches for designing an algorithm to solve the gym's timetable problem and make a recommendation to the gym as to which approach they should use. To do this:

• Describe how the two approaches could be applied to the problem of allocating gym classes to timetable blocks and discuss their comparative advantages and limitations.

• Explain how variables such as the number of programs, the typical number of programs enrolled in by each gym member, and the number of timetable blocks would influence your choice of method. Quantify time complexities where appropriate. 10 marks

END OF QUESTION AND ANSWER BOOK