SUPERVISOR TO ATTACH
PROCESSING LABEL HERE

Write your **student number** in the boxes above. **Letter**

Algorithmics (HESS)

Question and Answer Book

VCE Examination – Thursday 13 November 2025

- Reading time is **15 minutes**: 11.45 am to 12 noon
- Writing time is **2 hours**: 12 noon to 2.00 pm

Approved materials

- One scientific calculator

Materials supplied

- Question and Answer Book of 36 pages
- Multiple-Choice Answer Sheet

Instructions

- Follow the instructions on your Multiple-Choice Answer Sheet.
- At the end of the examination, place your Multiple-Choice Answer Sheet inside the front cover of this book.

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

Contents

VICTORIAN CURRICULUM AND ASSESSMENT AUTHORITY

Section A – Multiple-choice questions

Instructions

- Answer **all** questions in pencil on your Multiple-Choice Answer Sheet.
- 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.

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

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

Question 1

Eshal would like to store a collection of data items in a way that any item can be easily accessed.
Which abstract data type (ADT) would be most suitable for her to use?

A. array
B. graph
C. set
D. stack

Question 2

Which one of the following is the signature specification of the enqueue operation of a queue ADT?

A. queue → queue × item
B. queue × item → item
C. queue × item → queue
D. queue × item × item → queue

Do not write in this area.

Question 3

Which one of the following algorithm design patterns relies heavily on queues as an ADT?

A. breadth-first search

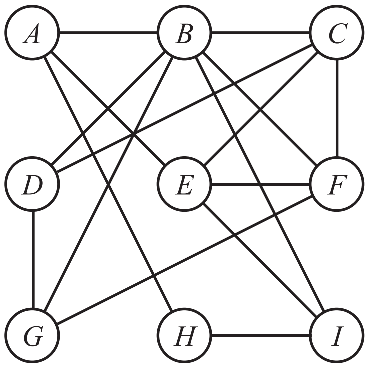
B. depth-first search

C. divide and conquer

D. dynamic programming

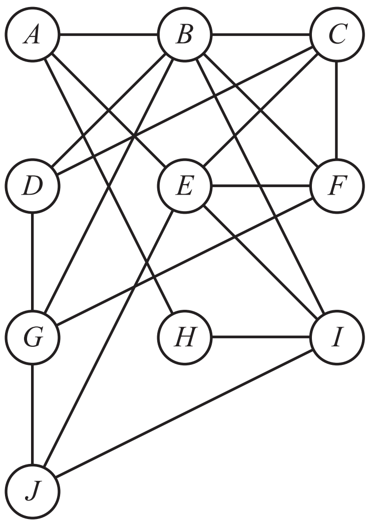
Question 4

Consider the following graph.

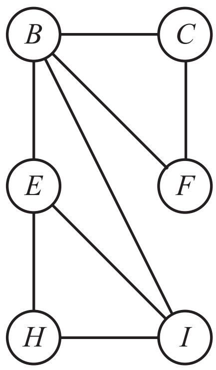


Which one of the following graphs is a subgraph of the graph above?

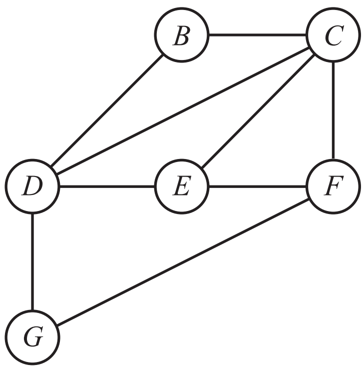
A.



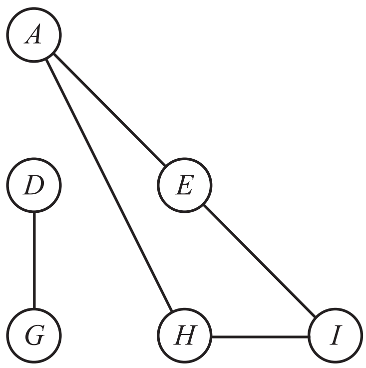
B.



C.



D.



Question 5

In which one of the following situations would the construction of a decision tree be justified?

A. A decision needs to be made based on a fixed set of conditions and input data.

B. Many decisions need to be made based on a fixed set of conditions and varying input data.

C. A decision needs to be made based on varying conditions and fixed input data.

D. Many decisions need to be made based on varying conditions and fixed input data.

Question 6

Which one of the following best describes Prim's algorithm for finding the minimum spanning tree in a weighted graph?

A. Explore all nodes at the present depth level before moving on to nodes at the next depth level.

B. Iteratively grow a spanning tree by adding the smallest edge connecting the tree to a vertex not yet in the tree.

C. Perform a depth-first search to determine the order of nodes for the construction of a spanning tree.

D. Utilise dynamic programming to minimise the total weight of the spanning tree.

Question 7

A freight delivery service for a warehousing company delivers all shipments from a central warehouse to depots in key suburbs. The company has one truck to deliver to all depots and the truck is only big enough to carry items for delivery to a single depot at a time.

If the company wants to find the fastest routes for its truck, which one of the following best describes its problem?

A. all-pairs shortest path

B. minimal spanning tree

C. single-source shortest path

D. travelling salesman

Question 8

The logistics team of a package delivery company needs to fill a container truck with packages. Each package has a weight and a value associated with it. The container truck has a maximum weight capacity C. The goal is to maximise the total value of the packages in the container truck without exceeding this weight capacity.

What type of problem is this?

A. 0-1 knapsack

B. minimal spanning tree

C. graph colouring

D. travelling salesman

Question 9

Consider the following recursive function for calculating the factorial of a number n.

```
Algorithm recursiveFactorial(n):
  If n = 0 Do
    Return 1
  Else
    Return n × recursiveFactorial(n − 1)
```

Which one of the following is the iterative equivalent of the recursive function above?

A.
```
Algorithm iterativeFactorial(n):
  If n = 0 Do
    Return 1
  result ← 1
  For i in {1, ..., n} Do
    result ← result × i
  Return result
```

B.
```
Algorithm iterativeFactorial(n):
  If n = 0 Do
    Return 1
  result ← 0
  For i in {n, ..., 1} Do
    result ← result + i
  Return result
```

C.
```
Algorithm iterativeFactorial(n):
  result ← 1
  While n > 1 Do
    result ← result × (n − 1)
    n = n − 1
  Return result
```

D.
```
Algorithm iterativeFactorial(n):
  result = n
  For i in {n − 1, ..., 1} Do
    result ← result ÷ i
  Return result
```

Do not write in this area.

Question 10

An artificial intelligence system is being developed to assist medical professionals in diagnosing diseases based on a set of symptoms. The system needs to make a series of decisions by evaluating various symptoms and their combinations to arrive at the most likely diagnosis.

Which one of the following tools would **not** be appropriate for creating a computer-assisted diagnostic tool?

A. support vector machines
B. neural network
C. decision tree
D. Turing Test

Question 11

Consider the following list of 12 items.

5	1	12	8	4	10	2	9	6	7	11	3

The mergesort algorithm is executed on this list.

The list returned by the left call of the algorithm is

A.

1	2	3	4	5	6

B.

1	4	5	8	10	12

C.

1	5	10	4	8	12

D.

1	5	12	4	8	10

Question 12

Consider solving a logic puzzle using a backtracking algorithm.

Which ADT is most appropriate to efficiently manage the puzzle's state and backtrack when needed?

A. array
B. queue
C. stack
D. linked list

Question 13

A robotic vacuum cleaner needs to navigate through a grid-based map of a house to reach the charging station from its current position. The grid contains obstacles such as walls and furniture.

In developing a navigation system, which one of the following algorithms is most suitable for efficiently finding the shortest path to the charging station while avoiding obstacles?

A. A* algorithm

B. breadth-first search

C. depth-first search

D. Dijkstra's algorithm

Question 14

In the Entscheidungsproblem, Hilbert and Ackermann posed the question whether an algorithm can decide that a given statement is provable using the rules of logic.

Which one of the following statements is true?

A. The answer to the Entscheidungsproblem is sometimes positive and sometimes negative, depending on the input and algorithm.

B. Alan Turing proved that there exists a Turing machine that can solve the Halting Problem, showing that the answer to the Entscheidungsproblem is negative.

C. Alonzo Church built on Alan Turing's proof by showing that lambda calculus expressions and Turing machines are equivalent, developing a subsequent response to the Entscheidungsproblem.

D. Alonzo Church proved that there is no computation function that can determine whether two lambda calculus expressions are equivalent, showing that the answer to the Entscheidungsproblem is negative.

Question 15

Jing is on holiday in a city. He is planning a daytrip that will start at his hotel and make stops at several tourist attractions. Jing wishes to find the shortest travel distance possible.

How does this problem differ from the ideal travelling salesman problem?

A. Not every destination has to be visited.

B. Destinations can be visited more than once.

C. The starting location does not need to be returned to.

D. The distance between destinations may change.

Question 16

Which one of the following is a correct set-up for the Turing Test?

A. Two artificial intelligence candidates are questioned by a human interrogator to determine which is more similar in behaviour to a human.

B. An artificial intelligence candidate and a human player are questioned by a human interrogator who must determine which is human and which is not.

C. Two humans are questioned by an artificial intelligence candidate. The humans must determine whether the candidate is human.

D. An artificial intelligence candidate and a human player are questioned by a human interrogator who must determine whether the computer or the human is more intelligent.

Question 17

Consider the following recurrence relationship.

$$f(n) = \begin{cases} f\left(\frac{n}{2}\right) + n^2 - 2n, & \text{if } n > 0 \\ 1, & \text{if } n = 0 \end{cases}$$

Applying the Master Theorem, the solution for $f(n)$ is

A. $O(1)$

B. $O(n)$

C. $O(n\log(n))$

D. $O(n^2)$

Question 18

What is the correct sequence of steps needed to train an algorithm with data?

A. Optimise the algorithm's parameters based on all instances of the data. Select a subset of the data for testing to evaluate the performance of the algorithm and its parameters.

B. Set the algorithm's parameters based on human expertise and existing knowledge. Use the data to evaluate the performance of the algorithm and its parameters.

C. Split the data into train and test sets. Optimise the algorithm's parameters based on the data sample in the train set. Use the test set to evaluate the performance of the algorithm and its parameters.

D. Split the data into train and test sets. Optimise the algorithm's parameters based on the data sample in the train set. Use the test set to evaluate the performance of the algorithm and its parameters. Re-sample the train and test sets until evaluation on the test set is maximised.

Question 19

What is an advantage of analysing algorithms using Big-O notation?

A. It allows us to trade off space and time.

B. It provides a lower bound on the time and space trends.

C. It allows us to directly compare the actual time and space costs of algorithms.

D. It allows us to ignore constants and multipliers, and concentrate on the time and space trends of algorithms.

Question 20

Which one of the following statements describes an advantage of randomised search compared to hill climbing on a heuristic function?

A. Randomised search is guaranteed to find an optimal solution.

B. Randomised search is easier to implement and has fewer hyperparameters to tune.

C. Randomised search is more efficient and converges on solutions faster than hill climbing.

D. Randomised search can be applied to problems where the heuristic function is not differentiable.

End of Section A

Do not write in this area.

This page is blank.

Examination continues on the next page.

Section B

Instructions

- Answer **all** questions in the spaces provided.
- Write your responses in English.

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

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

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

Question 1 (2 marks)

Mahir is reading a book about programming. The book gives an example of an isEqual operation for the set abstract data type (ADT). The signature specification given for the operation is

set × set → boolean

Explain what this signature specification means for the isEqual operation.

Question 2 (2 marks)

A stack contains the following data.

top →	2
	13
	8
	5

The following sequence of operations is performed on the stack.

```
pop()
pop()
push(6)
pop()
push(2)
push(7)
pop()
```

What data does the stack contain after these operations?

Question 3 (4 marks)

Emile is documenting a range of IT processes. The software that he uses stores steps in a list. An example of a process is shown below.

> **Example process – Disposal of used hard drives**
>
> 1. If the hard drive will be reused within the company and was never used for sensitive data, then securely erase the disk and then store it for reuse.
> 2. If the hard drive will be reused and was used for sensitive data, then send it to a hard drive sanitisation company for certified data destruction and then store it for reuse.
> 3. If the hard drive will be disposed of, then send the disk to a data destruction company for physical destruction and recycling.
> 4. If the hard drive was used for sensitive data, then store the drive's Certificate of Destruction.

The IT processes that Emile is documenting typically have 5–20 steps.

a. Describe one advantage and one disadvantage of using a list to store processes like the example above. 2 marks

Advantage ______________________________

____

____

Disadvantage ______________________________

____

____

b. Identify an alternative data structure that could be used to store such processes and explain **one** advantage that it would have over a list. 2 marks

____

____

____

____

____

Question 4 (2 marks)

The following algorithm is executed to construct a graph.

```
Algorithm createGraph():
  Let G be an empty graph
  For i in {1, ..., 6} Do
    Add node i to G
  u ← 1
  v ← 3
  While u ≤ 5 Do
    While v ≤ 6 Do
      Add edge (u, v) to G
      v ← v + 2
    u ← u + 2
    v ← u + 1
  Return G
```

Draw the graph produced by this algorithm.

Question 5 (3 marks)

Partially completed pseudocode for the Bellman-Ford algorithm is shown below. The algorithm takes three inputs:

- V, the set of the graph's vertices
- E, the set of the graph's edges
- s, a source vertex

```
   Algorithm BellmanFord(V, E, s):
1    Let n_V be the length of V
2    Let distance be a list of length n_V
3    For i in [0, ..., n_V - 1] Do
4      distance[i] ← ____________________
5    distance[s] ← 0
6    Repeat n_V - 1 Times
7      Foreach edge in E as (u, v) Do
8        If ______________________________________ Do
9          distance[v] ← distance[u] + weight(u, v)
10   Foreach edge in E as (u, v) Do
11     If ______________________________________ Do
12       Return Error( ___________________________ )
13   Return distance
```

a. To what value should the distance list be initialised in line 4? 1 mark

b. What condition should be tested in line 8 and line 11? 1 mark

c. What is the nature of the error that should be reported in line 12? 1 mark

Question 6 (6 marks)

a. Mabel is analysing a naive algorithm for calculating the nth Fibonacci number. She analyses the algorithm and estimates that its running time is bound by $O(2^n)$.
She concludes that the algorithm will not be useful for finding Fibonacci numbers for large values of n.

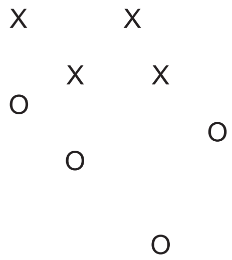
Her friend Sophie does more research and determines that the running time for the algorithm can be bound by $O(1.618^n)$.

Explain how Sophie's improved understanding of the algorithm's running time should influence Mabel's conclusion that the algorithm will not be useful for finding Fibonacci numbers for large values of n. 2 marks

b. Write pseudocode for a dynamic programming algorithm for finding the nth Fibonacci number that runs in linear time. 4 marks

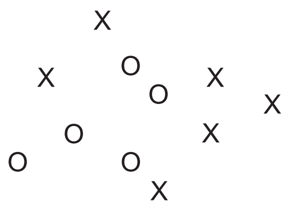
Question 7 (2 marks)

a. Consider the diagram below, showing eight points on a two-dimensional plane.



On the diagram above, draw a possible boundary for a linear classifier that perfectly separates the points labelled X from the points labelled O. 1 mark

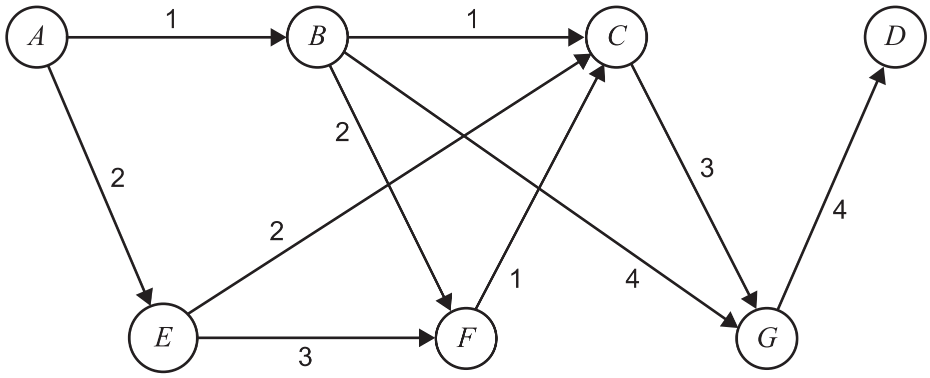
b. Consider the diagram below, showing 11 points on a two-dimensional plane.



Explain why a linear classifier can never perfectly separate the points labelled X from the points labelled O. 1 mark

Question 8 (4 marks)

Consider the following graph.



a. Using Dijkstra's algorithm, determine the shortest distance from node A to each other node and complete the table below. 1 mark

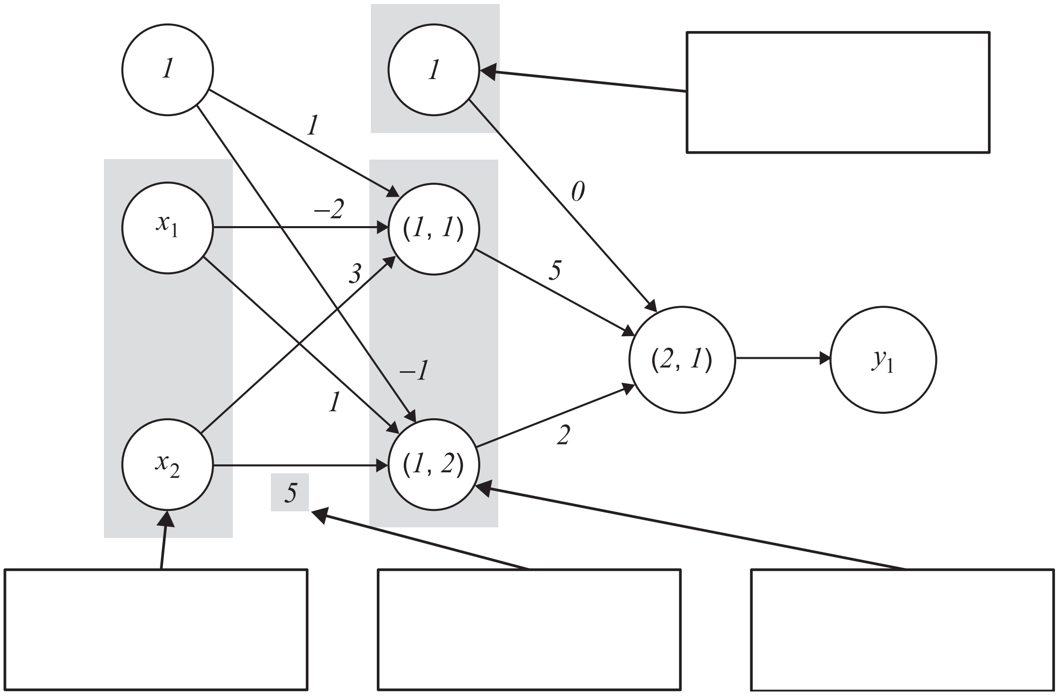
Node	A	B	C	D	E	F	G
Shortest distance							

b. The PageRank algorithm is applied to the graph above with a damping factor of 0.8. Each node is initialised with a PageRank of $\frac{1}{7}$. The algorithm is implemented in such a way that nodes with an out-degree of 0 distribute their rank equally among all other nodes.

Calculate the PageRank of node F after one iteration of the algorithm. 3 marks

Question 9 (4 marks)

Consider the following annotated diagram of a neural network.



a. Using one term from the list below for each box, complete the four blank boxes in the diagram above to annotate the parts of the neural network. 2 marks

input layer	output layer	hidden layer
backpropagation layer	transfer function	activation function
summation function	neural node	bias node
leaf node	support vector	edge weight

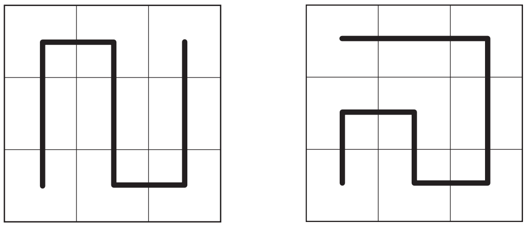
b. Nodes $(1, 1)$, $(1, 2)$ and $(2, 1)$ use a rectified linear transfer function, where the node output, $o(a)$, for a given activation, a, is given by

$$o(a) = \begin{cases} 0, & a < 0 \\ a, & a \geq 0 \end{cases}$$

Calculate the output value, y_1, for the input $x_1 = 3$ and $x_2 = 1$. 2 marks

Question 10 (4 marks)

In a space-filling walk puzzle, the solver is given a two-dimensional square grid and they are only allowed to make orthogonal moves in the grid. A space-filling walk of the grid is any path that visits every square in the grid exactly once. Two examples of a 3 × 3 grid are shown below.

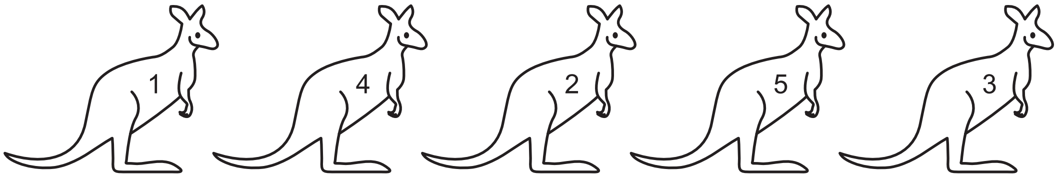


The time required to determine the unvisited grid locations adjacent to a particular grid location will significantly impact the running time of algorithms for solving these puzzles.

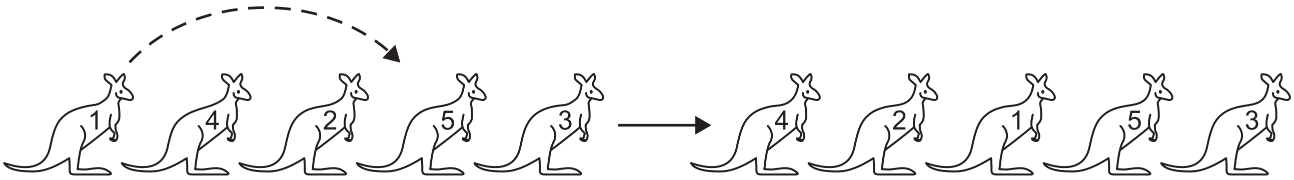
Describe how complete and partially complete space-filling walks could be stored using one or more ADTs, and how a set of unvisited grid locations adjacent to a particular grid location would be quickly determined from this data model.

Question 11 (4 marks)

In the game Queue Queue Kangaroo, five kangaroo figurines numbered 1 to 5 are thrown randomly onto the table and placed in a line.



The game requires that the kangaroos be rearranged so that they are in order – 1, 2, 3, 4, 5. Kangaroos love to hop, though, and the only move allowed is for one kangaroo to hop over two other kangaroos. Kangaroos can hop in either direction. One example of a move is shown below.



before move after move

a. Describe an ADT, or combination of ADTs, that could be used to store all of the arrangements of kangaroos and store which combinations can be reached from each other. 2 marks

b. Grace is playing a game of Queue Queue Kangaroo and is having trouble solving the game. She wonders whether the game is even solvable.

Describe an algorithm that could be used to determine whether the game can be solved from any possible starting point. 2 marks

Question 12 (3 marks)

Aachi has a program that tracks events she has in her calendar. The events are stored in a list. When Aachi adds a new event, it is added to the end of the list. Information about the event is stored as a single text string that includes all the relevant event information.

part of Aachi's list

Swimming 11/10/2025
School formal 5/9/2025
Camping 29/9/2025
Camping 30/9/2025
Swimming 18/10/2025

Aachi is considering the following upgrade options that will modify how she stores the event data:

1. Sort the list by event date and maintain the sort as each new event is added.
2. Convert the list to a dictionary, with each event stored as a (date, description) pair.
3. Split the text string into a description and a date, and store these in two separate lists.

Which of these options would have the biggest impact on reducing the time required to determine whether a new event clashes with an existing event? Justify your selection.

Question 13 (6 marks)

Jorge is playing a game that requires making words from a given list of tiles. Each tile has one letter printed on it and tiles may be arranged in any order. At any point in the game, Jorge may have anywhere from 1–10 tiles. The goal of the game is to create a word using all the tiles.

For example, the five tiles below could be rearranged to create any one of several possible words.

A	E	R	S	T

list of tiles

S	T	A	R	E

a possible word

Jorge is keen to develop an AI-like computer player that will generate words from a given set of tiles.

Jorge has a set, W, that contains many words. He has a second set, S, that contains all of the word stems of the words in W. A word stem is a sub-string found at the beginning of a word.

For example, the word stems of ‘stare’ are:

- ‘’ (the empty string)
- ‘s’
- ‘st’
- ‘sta’
- ‘star’

Jorge also has a list of tiles $T = [t_1, t_2, \ldots, t_n]$, where $n \geq 1$ and t_k is the letter written on tile k.

Using the backtracking algorithm design pattern, write pseudocode for an algorithm `findWord` that takes the following four inputs:

- W, a set of words
- S, a set of word stems for all words in W
- T, a list of tiles
- C, the current word or partial word, initially an empty string

This algorithm will return a word that can be formed using all n tiles or an empty string if no such word exists in W.

Question 14 (4 marks)

Let *text* be a list of lists, where the sub-lists contain lower-case strings containing no spaces or punctuation. An example is shown below.

$$SampleText = \begin{bmatrix} [\text{'}it\text{'}\ \ \text{'}was\text{'}\ \ \text{'}hot\text{'}] & \\ [\text{'}the\text{'}\ \ \text{'}dog\text{'}\ \ \text{'}lay\text{'}\ \ \text{'}down\text{'}] & \\ [\text{'}dog\text{'}\ \ \text{'}done\text{'}] & \end{bmatrix}$$

Let *read* and *write* be strings containing no spaces or punctuation.

The algorithm *FindAndReplace* takes three inputs, *text*, *read* and *write*, and returns an updated text. The algorithm with one sub-algorithm is described below.

Note: Comparing the equality of two strings takes time proportional to the length of the strings.

```
Algorithm FindAndReplace(text, read, write):
  For j in [1, ..., len(text)] Do
    sentence ← text[j]
    idx ← Search(sentence, read)
    If idx > 0 Do
      sentence[idx] ← write

Algorithm Search(A, read):
  For i in [1, ..., len(A)] Do
    If A[i] = read Do
      Return i
  Return -1
```

Analyse the pseudocode to find the Big-O time complexity of *FindAndReplace*. Define relevant variables and explain how you derived your solution.

Question 15 (6 marks)

The algorithm *ClosestPair* takes one input, p, which is a set of n points in one-dimensional space. The algorithm returns the distance between the closest pair of points.

In the following pseudocode:

- `abs` denotes the absolute value function, equivalent to $abs(x) = \begin{cases} -x, & x < 0 \\ x, & x \geq 0 \end{cases}$, which runs in constant time
- `floor` denotes the floor function, which rounds down a number to its nearest whole number and runs in constant time
- `p[a, ..., b]` denotes a sub-list of p containing all values in p between indexes a and b inclusive. Creating such a sub-list takes time $O(b - a)$.

```
Algorithm ClosestPair(p):
  n ← length(p)
  If n < 2 Do
    Return +∞
  If n = 2 Do
    Return abs(p[0] - p[1])
  mid ← floor(n/2)
  pLeft ← p[0, ..., mid - 1]
  pRight ← p[mid, ..., n - 1]
  minLeft ← ClosestPair(pLeft)
  minRight ← ClosestPair(pRight)
  minCross ← CrossCheckPairs(pLeft, pRight)
  Return min(minLeft, minRight, minCross)

Algorithm CrossCheckPairs(pLeft, pRight):
  min ← +∞
  For l in pLeft Do
    For r in pRight Do
      If abs(l - r) < min Do
        min ← abs(l - r)
  Return min
```

a. Let n_L and n_R denote the length of *pLeft* and *pRight* respectively. Let $A(n_L, n_R)$ denote the running time of `CheckCrossPairs` in terms of n_L and n_R.

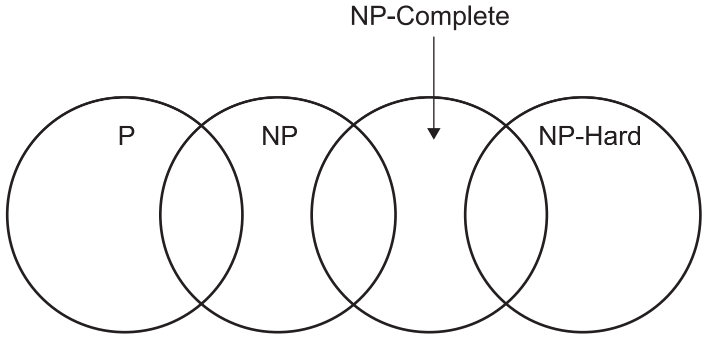
Analyse the function `CheckCrossPairs` to find its Big-O time complexity for $A(n_L, n_R)$ with respect to n_L and n_R. As part of your response, annotate the pseudocode on page 26 to show how you derived this time complexity. 2 marks

b. Let n be the length of p. Let $B(n)$ denote the running time of `ClosestPair` in terms of n.

Write a recurrence relation for $B(n)$, including the base case. Hence, derive the running time of `ClosestPair` using the Master Theorem. 4 marks

Question 16 (4 marks)

Consider the following Venn diagram of the space of computational problems.



Based on your understanding of the definitions of P, NP, NP-Hard and NP-Complete, explain how the diagram is incorrect and draw a correct diagram in the box below. Support your response with relevant definitions of these problem classes.

Question 17 (5 marks)

The systems response to Searle's Chinese Room Argument states that even though the human agent may not understand Chinese, the system as a whole exhibits behaviours equivalent to understanding Chinese and hence this invalidates Searle's argument.

a. Describe **one** counterargument to the systems response. 2 marks

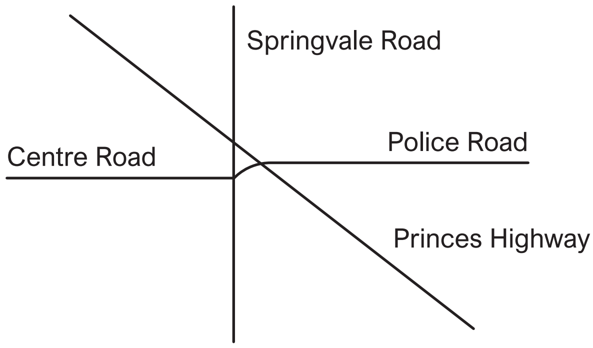
b. Consider the following statement:

'The systems response is the only valid counterargument to Searle's Chinese Room Argument.'

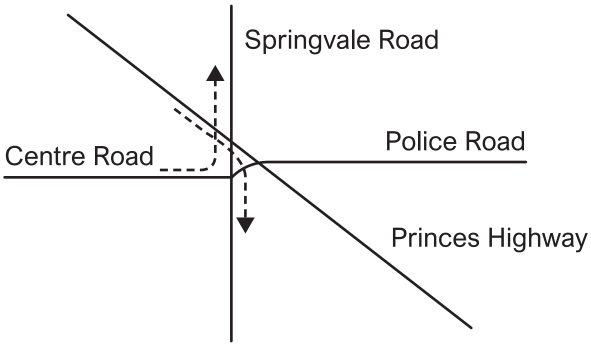
Do you agree or disagree with this statement? Justify your response. 3 marks

Question 18 (5 marks)

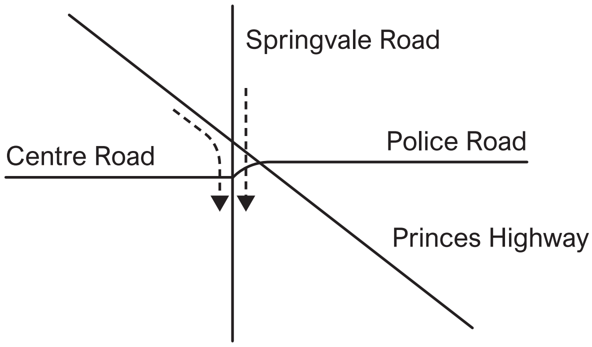
Springvale Junction is an intersection in south-eastern Melbourne where four roads meet. The current layout of the intersection is shown in the diagram below.



Traffic through the intersection is controlled by traffic lights. The traffic lights cycle through several sets of lights that allow non-intersecting paths of traffic to travel through the intersection. Two traffic paths intersect if either the traffic paths cross each other or if they exit from the intersection via the same road. Two examples of intersecting paths are shown below.



Two paths crossing each other



Two paths exiting via the same road

Describe an algorithmic method for finding the minimal number of sets of non-intersecting traffic paths that need to be cycled through to allow all traffic to travel through the intersection.

Support your response by describing

- how a graph would be used to model the intersection and its traffic paths
- a suitable algorithmic approach for determining sets of non-intersecting traffic paths that the traffic lights could cycle through.

Question 19 (10 marks)

An examination scheduler needs to develop a seating plan for several examination sessions. More than one examination may be scheduled to run at the same time within a session. All examination sessions are held in a large hall with 30 rows of 13 seats. One of the rules is that no two students sitting the same examination may be seated adjacent to each other, either to the left, right, front or behind.

a. Consider the following special case:

- The number of subjects, n, is greater than 1 and less than 13.
- There is an equal number of students for all subjects. Let this be denoted by s.
- The total number of students is at most $30 \times 13 = 390$.

Let N be a list of lists, with $N[j]$ being the list of names of the students sitting subject j and $N[j][i]$ being the name of the ith student for subject j.

Consider the following algorithm that creates a seating plan for this special case.

```
Algorithm assignSeats(N, n, s):
  Let seats be a 30 by 13 array
  x ← 0, y ← 0
  For i in [0, ..., s - 1] Do
    For j in [0, ..., n - 1] Do
      seats[x][y] ← N[j][i]
      x ← x + 1
      If x ≥ 13 Do
        y ← y + 1
        x ← 0
  Return seats
```

Demonstrate that this algorithm produces a valid seating plan using an argument by contradiction. 3 marks

b. Consider now the more general case in which there may be different numbers of students completing each subject.

Describe a suitable method for determining whether a particular set of subjects can be scheduled for the same examination session. To do this:

- define the relevant variables
- describe how the information about the problem should be represented
- describe an algorithmic approach for determining whether a seating plan could be developed.

7 marks

End of examination questions

This page is blank.

This page is blank.