

STUDENT NUMBER           Letter

## ALGORITHMICS (HESS)

### Written examination

Day Date

Reading time: \*.\*.\* to \*.\*.\* (15 minutes)

Writing time: \*.\*.\* to \*.\*.\* (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

### 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.

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

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

An abstract data type (ADT) has the following partial signature specification.

$$\text{Operation\_1: } \textit{element} \times Y \rightarrow Y$$

The ADT  $Y$  could be

- A. a list ADT, where Operation\_1 is the ‘add’ operation.
- B. an array ADT, where Operation\_1 is the ‘lookup’ operation.
- C. a queue ADT, where Operation\_1 is the ‘dequeue’ operation.
- D. a stack ADT, where Operation\_1 is the ‘pop’ operation.

#### Question 2

Consider the following algorithm.

```

Algorithm numberbox(n)
  a ← 1
  b ← 1
  For each digit in n Do
    If digit is even Do
      a ← a + b
      b ← b + 1
    If digit is odd Do
      a ← a × b
      b ← b - 1
  Return a
  
```

What does the algorithm numberbox output when  $n = 2023$ ?

- A. 3
- B. 11
- C. 12
- D. 28

DO NOT WRITE IN THIS AREA

**Question 3**

Which one of the following descriptions of an array is correct?

- A. An array is ordered alphabetically by key.
- B. An array can have an unlimited number of elements.
- C. An array allows only sequential access to its elements.
- D. An array allows access to each element based on the element's index.

**Question 4**

At one Victorian school, a student is allowed to enrol in advanced studies in a subject if they satisfy all of the following requirements:

- They are not studying a language outside of school.
- They are achieving high marks in the subject.
- They are not failing any other subjects.

Which one of the following conditions would be suitable for testing whether a student can study an advanced subject?

- A. NOT (studying a language outside of school AND failing other subjects) AND (getting high marks in the subject)
- B. NOT (studying a language outside of school OR failing other subjects) AND (getting high marks in the subject)
- C. NOT studying a language outside of school OR NOT failing other subjects OR getting high marks in the subject
- D. NOT (studying a language outside of school OR failing other subjects) OR (getting high marks in the subject)

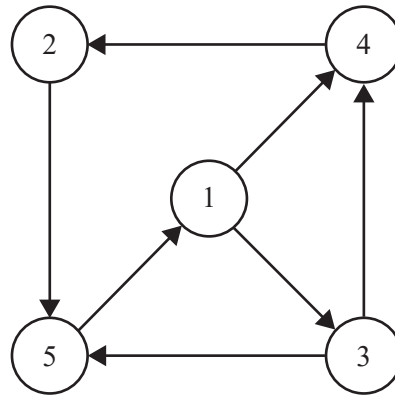
**Question 5**

The design pattern that best describes Prim's algorithm is

- A. brute force, as Prim's algorithm is required to check every edge in a graph.
- B. brute force, as Prim's algorithm exhaustively compares each potential spanning tree and chooses the minimum cost spanning tree.
- C. greedy, as Prim's algorithm expands its edge set by selecting the locally optimal edge at each step.
- D. divide and conquer, as Prim's algorithm divides the graph into two sets of vertices at each step, those connected and those not connected.

Use the following information to answer Questions 6 and 7.

Consider the following graph.



Hamed, an Algorithmics student, attempts to run the PageRank algorithm on the graph above using a damping factor of  $d = 0.85$ . He obtains the following results, correct to four decimal places, after the first two iterations.

Iteration	Node 1	Node 2	Node 3	Node 4	Node 5
0 (initialisation)	0.2000	0.2000	0.2000	0.2000	0.2000
1	0.2000	0.2000	0.1150	0.1639	0.2489
2	0.2257	0.1907	0.1259	0.1795	0.2456

### Question 6

It is certain that Hamed's results are **incorrect** because

- A. the PageRanks of Node 1 and Node 2 are unchanged after the first iteration but convergence should not happen until much later.
- B. the PageRank of Node 1 is increasing, yet Node 1 has more outbound links than incoming links.
- C. the PageRank algorithm needs to be amended in the presence of cycles.
- D.** the sum of PageRanks after the first iteration is less than 1.

### Question 7

The correct PageRank for Node 3 after two iterations is

- A.** 0.1150
- B. 0.1259
- C. 0.1457
- D. 0.1639



**Question 8**

Which one of the following graph algorithms repeatedly iterates over the set of all edges in its input?

- A. Bellman-Ford algorithm
- B. Dijkstra's algorithm
- C. Floyd's algorithm
- D. Floyd-Warshall algorithm

**Question 9**

Which one of the following statements is true?

- A. All connected graphs are trees.
- B. All trees are connected graphs.
- C. All connected graphs are cyclic graphs.
- D. All cyclic graphs are connected graphs.

**Question 10**

Which one of the following recurrence relations describes the worst case for the number of comparisons made by the quicksort algorithm on an input of  $n$  elements?

- A.  $T(n) = 2T\left(\frac{n}{2}\right) + O(n)$
- B.  $T(n) = 2T(n) + O(n)$
- C.  $T(n) = T(n-1) + O(n)$
- D.  $T(n) = T(n^2) + O(n)$

Use the following information to answer Questions 11 and 12.

Consider a country where the only coins in circulation are 1-cent, 3-cent and 4-cent coins. A citizen would like to determine the fewest coins required to pay for a chocolate bar. The following algorithm has been provided for this purpose. The last line of the algorithm, which makes three recursive calls to itself, is incomplete.

**Input:**  $x$ , the price of the chocolate bar given in cents

**Algorithm** minCoins( $x$ ):

**If**  $x < 5$  **Then**

**If**  $x = 2$  **Then**

**Return** 2

**Else**

**Return** 1

**Else**

**Return** \_\_\_\_\_

### Question 11

Which one of the following correctly completes the algorithm above?

- A.  $\text{minimum}(\text{minCoins}(x-1)+1, \text{minCoins}(x-3)+1, \text{minCoins}(x-4)+1)$
- B.  $\text{minimum}(\text{minCoins}(x+1)-1, \text{minCoins}(x+3)-1, \text{minCoins}(x+4)-1)$
- C.  $\text{minimum}(\text{minCoins}(x-1)-1, \text{minCoins}(x-1)-3, \text{minCoins}(x-1)-4)$
- D.  $\text{minimum}(\text{minCoins}(x-1)+1, \text{minCoins}(x-1)+3, \text{minCoins}(x-1)+4)$

### Question 12

Why is it infeasible to run this algorithm on large values of  $x$ ?

- A. This approach is greedy and would not return an optimal value when the value of  $x$  gets large.
- B. Recursive algorithms will necessarily take longer to execute than iterative algorithms.
- C. A very large number of repeated recursive calls would mean the algorithm is unlikely to terminate in a reasonable amount of time.
- D. The minimum function is likely to be slow, so calling it repeatedly will mean the algorithm is unlikely to terminate in a reasonable amount of time.

### Question 13

An algorithm has a double nested loop that compares each pair of items from a collection of  $n$  items.

This structure is characteristic of algorithms with a time complexity of

- A.  $O(1)$
- B.  $O(n)$
- C.  $O(n^2)$
- D.  $O(2^n)$

**Question 14**

Consider the recurrence relation

$$T(n) = \begin{cases} 5T\left(\frac{n}{2}\right) + 2n & \text{if } n > 1 \\ 2 & \text{if } n = 1 \end{cases}$$

$T(n)$  describes a function that is

- A.  $O(n)$
- B.  $O(n \log n)$
- C.  $O(n^{\log_2 5})$**
- D.  $O(n^2)$

**Question 15**

Which one of the following is a characteristic of the instruction table of a Turing machine?

- A. There must be instructions for every state of the Turing machine.
- B. The instructions for each state, other than the halt state, must include conditions for each possible symbol.**
- C. The instructions for each state, other than the halt state, must transition the Turing machine to a different new state.
- D. The instructions cannot direct that the Turing machine write a ‘blank’ symbol.

*Use the following information to answer Questions 16 and 17.*

A school is using the following combination of ADTs to store information about its classes:

- a dictionary in which each key is a teacher’s name and each value is an array of classes that the teacher teaches

$$\left\{ \begin{array}{l} \text{Ms Algo : [ '7 English A', '9 History C', '12 English D' ]} \\ \text{Mr Rithmics : [ '8 Maths B', '10 Maths C', '12 Further Maths E' ]} \end{array} \right\}$$

- an associated dictionary that stores the names of the students in each class

$$\left\{ \begin{array}{l} \text{'7 English A': [ 'Robert Prim', 'Lester Ford', ... ]} \\ \text{'8 Maths B': [ 'Richard Bellman', 'Edsger Dijkstra' ... ]} \end{array} \right\}$$

**Question 16**

Which one of the following operations could **not** be performed in  $O(1)$  time?

- A. Set the first item in the array for ‘Ms Algo’ to ‘7 English D’.
- B. Determine whether a given teacher teaches at the school.
- C. Determine which maths class a given student is enrolled in.**
- D. Determine if there is a class called ‘7 History G’ at the school.

**Question 17**

Let  $c$  be the number of classes in the school and let  $s$  be the number of students in the school. An algorithm iterates over each class to find the number of classes with 25 or more students. The time complexity of this algorithm is

- A.**  $O(c)$
- B.  $O(s)$
- C.  $O(c + s)$
- D.  $O(c \times s)$

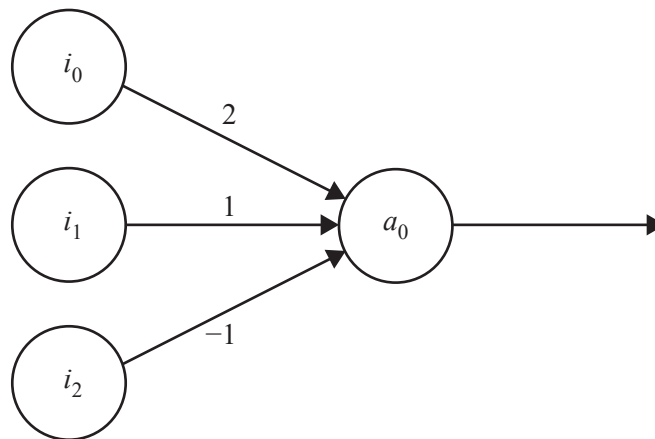
**Question 18**

Which of the following statements describes an aspect of Church's and Turing's resolutions to the *Entscheidungsproblem*?

- A. They defined what constituted a valid mathematical statement.
- B.** They defined what constituted a valid method for proving the truth of mathematical statements.
- C. They showed how all mathematical statements could be shown to be either true or false.
- D. They showed that more axioms were required to be able to prove the truth of mathematical statements.

**Question 19**

A small multi-layer perceptron neural network is shown below.



What is the activation of node  $a_0$  when the input nodes  $i_0$ ,  $i_1$  and  $i_2$  are set to output 1, 0 and 0.5 respectively?

- A. 1
- B.** 1.5
- C. 2
- D. 2.5

**Question 20**

The A\* algorithm attempts to minimise  $f(n) = g(n) + h(n)$ , where  $h(n)$  is a heuristic for the distance between node  $n$  and the goal.

For the algorithm to always find the shortest path, the function  $h(n)$  must

- A. evaluate to the true cost from  $n$  to the goal.
- B. not underestimate the true cost from  $n$  to the goal.
- C. overestimate the true cost from  $n$  to the goal.
- D.** not overestimate the true cost from  $n$  to the goal.

DO NOT WRITE IN THIS AREA

## SECTION B

## Instructions for Section B

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

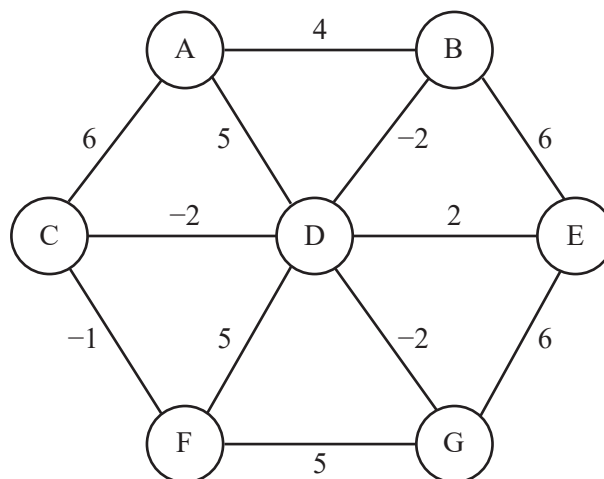
Use the Master Theorem to solve recurrence relations of the form shown below.

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

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

## Question 1 (6 marks)

Consider the following graph with edge weights as shown.



- a. Write a definition for the graph abstract data type (ADT).

2 marks

$$V = \{A, B, C, D, E, F, G\} \text{ nodes}$$

$$E = \{A-D, A-B, A-C, B-D, B-E, C-D, C-F, D-E, D-F, D-G, E-G, F-G\}$$

$$G = \{V, E\} \quad |V| = 7, |E| = 12$$

- b. What is the distance of the C-D-E-G path?

1 mark

6

c. Consider running Dijkstra’s algorithm, starting from node A, to find the shortest distance to all other nodes.

i. What property of this graph would make Dijkstra’s algorithm unsuitable?

1 mark

negative weighted edges

ii. Find a node for which the distance from node A that Dijkstra’s algorithm returns is incorrect. State the distance found by Dijkstra’s algorithm and the true shortest distance, respectively, in your response.

2 marks

for A-B Dijkstra's gives 4 the  
actual shortest length is 2  
follow A → C → D → B  
6        -2        -2

DO NOT WRITE IN THIS AREA

**Question 2** (5 marks)

Alex is developing a computer simulation to model a chemical reaction. In the simulation, the reactor is initially empty and the reaction starts when one or more chemicals are added. More chemicals may be added while the reaction is in progress. A chemical reaction is characterised by what chemicals are added, how much of them are added and when they are added.

- a. To store a description of a chemical reaction for simulation, Alex needs the following information about each addition into the reactor:
- the name of the chemical added
  - the time the chemical was added, in seconds, after the reaction begins
  - the amount of the chemical added, in milligrams

Describe a combination of ADTs that Alex could use to store a description of a chemical reaction. Justify your answer.

3 marks

dictionary chem Events (key = time,  
 values = { [chem 1, mass], [chem 2, mass], ...  
 the values are a list of arrays where  
 array chemical [symbol, mass mg]  
 time is important for reactions and is  
 unique and sequential for identifying when  
 chemicals have been added with the  
 recorded mass.

- b. Alex intends to use the combination of ADTs described in **part a.** often and decides to define the combination of ADTs as a new ADT called ChemEvents.

For the ChemEvents ADT, as described in **part a.**, write the signature specification of the following operations:

- Add a new chemical to the reaction.
- Look up what chemical was added to the reaction at a given time.

2 marks

addNewChemical (time, { [symbol, mass], [symbol, mass] }  
 ChemEvents × key × List → ChemEvents  
 lookupChemicals (time) dictionary ChemEvents  
 ChemEvents × key → List  
 { [s, m], [s, m] ..... }



**Question 3** (10 marks)

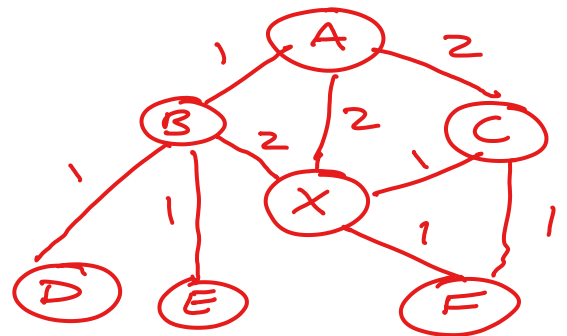
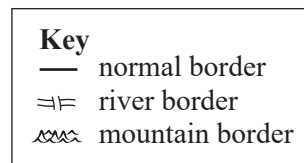
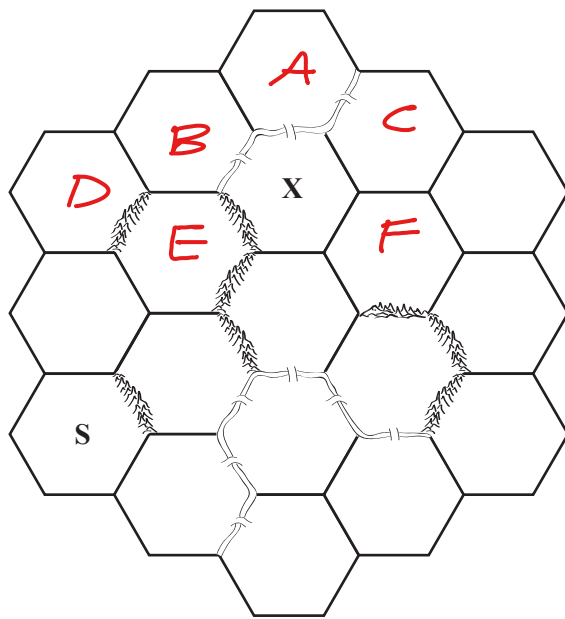
HexaQuesta is a game played on a board that is divided into hexagonal regions. There are three kinds of borders between regions:

- Normal borders take one move to cross.
- River borders take two moves to cross.
- Mountain borders cannot be crossed.

Game boards may contain thousands of hexagonal regions.

The objective of the game is to travel across the board from the start, **S**, to the finish, **X**, with the fewest number of moves possible.

An example of a HexaQuesta board, showing the start, S, and the finish, X



Samarth would like to write a program that determines how to win the game.

- a. Describe an ADT that appropriately models a board of the HexaQuesta game. 2 marks

Undirected graph of nodes and edges  
where a node is a hexagonal region  
and edges represent reachable adjacent  
regions, the edges have weights showing  
cost to travel to neighbour

- 1 normal neighbour
- 2 river crossing neighbour
- no edge indicates regions are separated by mountains or not neighbours

DO NOT WRITE IN THIS AREA

- b. Write an algorithm, in pseudocode, that Samarth could use to efficiently find a winning path from the start to the finish on any board.

5 marks

Dijkstras ( $G, S, T$ )<sup>start target</sup>

```

foreach node in G
  node.distance = ∞
  node.previous = ?
  node.visited = false
S.distance = 0
while (unvisited nodes exist) OR
  (T.visited = false) do
  V = unvisited node with smallest distance
  foreach neighbour U of V do
    if (U.distance >
      V.distance + weight(U-V)) then
      U.distance = V.distance + w(U-V)
      U.previous = V
    endif
  V.visited = true
  end do
end do
if (T.visited = true) then
  return (Constructpath(S, T))
else
  return (no path)
endif
end algorithm

```

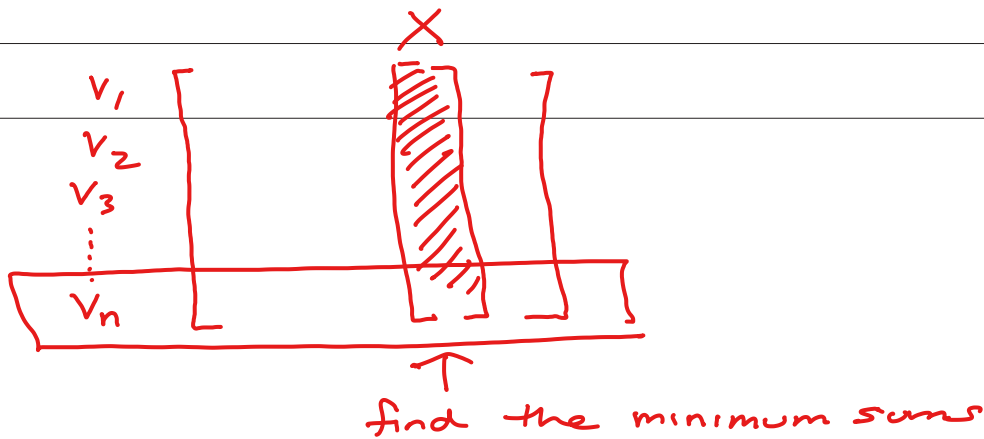
- c. Samarth would like to find the best starting region on a given board. He decides that the best starting region is the one with the lowest average number of moves to travel to every other region on the board.

Describe how Samarth could efficiently find the best starting region on the board and derive a running time for your solution. Do **not** provide pseudocode in your answer.

3 marks

Run Floyd-Warshall  $O(V^3)$  for all pairs shortest paths.

Use the rows of the resulting distance matrix to find the nodes with the lowest average number of moves to other nodes to determine starting node.



DO NOT WRITE IN THIS AREA

**Question 4** (3 marks)

Describe the characteristics of problems in the P time complexity class and give an example of **one** problem in this class.

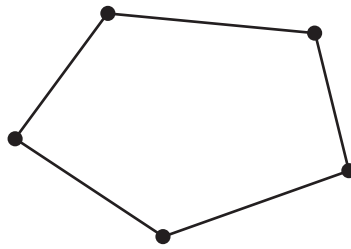
Problems in P class have fastest  
proven correct algorithms that run  
in polynomial time.

The sorting problem is P class as  
the Merge Sort algorithm solves it  
correctly in  $O(n \log n)$  time

which is  $O(n) < O(n \log n) < O(n^2)$   
between linear and quadratic — loglinear  
time.

**Question 5** (3 marks)

A ring graph is a graph with at least three nodes, in which a single cycle exists containing all vertices. An example of a ring graph with five vertices is shown below.



Consider the algorithm shown below that takes one integer input  $n \geq 3$ .

```

Algorithm RingGraphMaker(n) :
1   If n = 3 Then
2     Return a complete graph with three nodes
3   Else
4     G = RingGraphMaker(n-1)
5     Add a new node u to G
6     Randomly select an edge (p,q) in G
7     Create the edges (p,u) and (q,u) in G
8     Delete the edge (p,q)
9     Return G
10  EndIf

```

Demonstrate the correctness of the RingGraphMaker algorithm for  $n \geq 3$ .

*Base case:  $n=3$  when  $n=3$  the algorithm correctly returns a ring graph as graph exists*

*$n=2$  or  $n=1$  are not valid inputs since the correctness is for  $n \geq 3$*

*Inductive step: assume algorithm is correct for  $k$  nodes where  $k \geq 3$  and therefore show correct for  $k+1$  nodes (ie all  $n \geq 3$ )*

*Ringmaker( $k+1$ ) creates  $G = \text{Ringmaker}(k)$  which is known to be correct to the base case  $n=3$  which results in termination of recursion. on return from the recursion it adds a new node to graph  $G$  randomly selects an edge  $p-q$  and "breaks" it to include the new node  $u$  resulting in  $p-u-q$  which maintains the connectedness and the ring pattern of the graph  $G$ . Hence terminating correctly for  $k+1$  nodes.*

**Question 6** (10 marks)

TeleG, a mobile service provider, is building mobile phone towers for its new network along a straight, regional road that ends at a lighthouse. The road is more than 1000 km long and the houses along this road are spread out at irregular intervals. There are  $h$  houses in total and the location of each is described by its distance from the lighthouse at the end of the road. The distances to the lighthouse of all  $h$  houses are stored in an ordered array.

For example, an array of [1, 5, 27] would indicate that there are houses at 1 km, 5 km and 27 km from the lighthouse.



TeleG would like to build as few towers as possible along the road, while still providing mobile phone reception to every house. Towers can be built anywhere along the road  $k$  kilometres from the lighthouse, where  $k$  is an integer. Each tower provides mobile phone reception to all houses within  $r$  kilometres in either direction along the road.

- a. Describe a brute-force algorithm that would solve the problem of finding the smallest number of towers required to provide reception to all houses.

3 marks

Try a tower in each location or not for an  $n$  kilometre road.

0--1--2--3--4--.....k--k+1--.....n

This generates  $2^n$  combinations then check each combination for the cheapest combination. This is a similar problem to the 0/1 knapsack problem that has capacity  $n$ , put a tower in a location or not.

- b. Explain whether the brute-force algorithm described in **part a.** is feasible. Do **not** attempt to derive an exact time complexity.

2 marks

Generate and test brute force has exponential  $2^n$  combinations to test, where  $n$  is the length of the road in km and towers placed at integer intervals.

This is not feasible as  $n$  approaches very large values.

An alternative approach involves installing towers one by one, exactly  $r$  kilometres further from the lighthouse than the next house currently without mobile phone reception, until all houses are provided with reception.

For example, if the position of the houses was given by the array [3, 8, 11, 13] and  $r = 2$ , the towers would be installed as follows:

- The first tower would be installed 2 km further from the lighthouse than the nearest house, so it would be installed 5 km away from the lighthouse.
- The next house without reception is 8 km away from the lighthouse, so the second tower would be installed 10 km away from the lighthouse.
- The next house without reception is 13 km away from the lighthouse, so the third tower would be installed 15 km away from the lighthouse.



- c. State the algorithm design pattern that this approach uses. 1 mark

**Greedy, decrease & conquer approach.**

- d. Write an algorithm, in pseudocode, for the approach stated in **part c**. Your pseudocode should:

- take as input two parameters – the first an array named `houses` and the second an integer  $r$
- return the smallest number of towers required to provide each house along the road with mobile phone reception.

4 marks

```

Algorithm PlaceTowers(houses, r)
  create empty list towerList
  for h = 1 to length(houses) do
    if length(towers) > 0 then
      foreach tp in towerList do
        if (|houses[h]-tp| > r) then
          add(houses[h]+r) to towerList
        endif
      enddo
    else
      add(houses[h]+r) to towerList
    endif
  end do
  return length(towerList)
end Algorithm
  
```

**Question 7** (8 marks)

A tennis club keeps information on its members in an array named `members`. The element at index  $i$  of the array is the number of matches played by Member  $i$ .

For example, `members = [12, 15, 9]` would indicate that Member 0 has played 12 matches, Member 1 has played 15 matches and Member 2 has played 9 matches.

The club would like to reward the member who has played the most matches and has decided to use the following algorithm to do this.

```

Algorithm MaxTennis(Lower, Upper)
Begin
  If Lower = Upper Then
    Return Lower
  Else
    Midpoint = Lower + floor((Upper - Lower)/2)
    Best1 = MaxTennis(Lower, Midpoint)
    Best2 = MaxTennis(Midpoint + 1, Upper)
    If members[Best1] > members[Best2] Then
      Return Best1
    Else
      Return Best2
    EndIf
  EndIf
End

```

- a. Identify the design pattern that the algorithm above uses.

1 mark

**Divide and Conquer**

- b.

$$T(n) = \begin{cases} 2T\left(\frac{n}{2}\right) + 1 & \text{if } n > 1 \\ 1 & \text{if } n = 1 \end{cases}$$

Assuming that the number of members of the club is a power of two and that comparison of two integers can be done in constant time, explain why the time complexity of the club's algorithm can be obtained using the recurrence relation above.

3 marks

There are 2 recursive calls  $\implies T(n) = 2T(\dots)$  and each recursive call splits the input into two halves indicated by  $T(n) = 2T(n/2) \dots$   
 The work done outside of the recursion is constant  $O(1)$  as  $T(n) = 2T(n/2) + 1$ , and the base case when  $n = 1$  does one action.



- c. Calculate the time complexity of the club's algorithm.

2 marks

$$a=2, b=2, c=0, \text{ use Master Theorem } a > b^c$$

---

$$O(n^{\log_b a}) = O(n^{\log_2 2}) = O(n)$$

---

- d. Compare the running time of the club's algorithm to that of a brute-force search of the members array for its maximum value.

2 marks

A brute force approach would linearly search the entire array of  $n$  elements and keep track of the best maximum so far. This is  $O(n)$ .

---

DO NOT WRITE IN THIS AREA

**Question 8** (10 marks)

The decision version of the travelling salesman problem asks: given a weighted, undirected graph and a source node, does there exist a path starting and finishing at the source node that visits all remaining nodes exactly once and has a total weight of less than  $X$  (where  $X$  is a given integer)?

- a. Explain why the decision version of the travelling salesman problem is in NP. 2 marks

This version is in NP since the correct solution can take  $O(n!)$  time to discover – non polynomial time, while a potential solution can be verified for correctness in polynomial time.

- b. Describe a greedy approach to solving the travelling salesman problem. 2 marks

The nearest neighbour heuristic is a greedy approach to solving the TSP. From any city pick the next closest city and repeat until returned to starting city

– other solutions could talk about Hill Climbing  
– part a) assumes part b is Hill Climbing?

- c. An alternative approach to solving the travelling salesman problem is to use a simulated annealing algorithm. The greedy approach described in **part b.** can be used to generate an initial candidate solution for the simulated annealing algorithm. High-level pseudocode for the simulated annealing algorithm is provided below.

```

Generate initial candidate solution, S
Repeat until a terminating condition is reached:
    Generate a new candidate solution, S_new
    Set a temperature, T
    If the new candidate solution is accepted:
        Set S = S_new
Return S

```

*assumes  
hill climbing  
heuristic  
-not nearest  
neighbour*

- i. Explain how the algorithm would determine whether a new candidate solution is accepted. 3 marks

*In simulated annealing new solutions that are better should always be accepted. While the temperature  $T$  is "hot" according to a probability  $e^{-\Delta/T}$  worse solutions can be accepted to try and escape local optima. As  $T$  cools down  $e^{-\Delta/T}$  will approach 0 and fewer bad moves will be accepted.*

- ii. State a terminating condition that could be used for this algorithm. 1 mark

*A terminating condition when the temperature  $T$  is less than or equal to some threshold boundary  $T_0$  ( $T \leq T_0$ )*

- iii. Describe **one** advantage and **one** limitation of using a simulated annealing algorithm to solve the travelling salesman problem. 2 marks

*Adv: runs quickly, tries to escape local optima*

*Disadv: can still get stuck on local optimums - solution space requires preparation to identify metrics of improvement on previous solutions.*

**Question 9** (7 marks)

The Two Bays Shire is a rural local council responsible for the maintenance of 3000 km of local roads. The council has asked for submissions from local businesses to maintain the roads. Each submission states how many kilometres of road the business will maintain and how much it will charge the council per year. More than a hundred businesses apply.

A summary of some of the submissions is shown in the table below.

Business	Length of road (km)	Cost (\$)
:	:	:
Ricky's Road Maintenance	230	2 300 000
Smith & Son RM Services	455	4 200 990
Alice's Deluxe Road Maintenance	148	3 600 999
:	:	:

The Chief Operating Officer of the council needs to determine which businesses should be awarded the work so that the cost for the council is minimised.

- a. Explain whether or not the problem of finding the optimal collection of businesses to maintain the roads is tractable.

3 marks

3000 km is a fixed capacity similar to knapsack or problem as item = business (in/out) weight = length of road, value = cost \$, choose subset of businesses for 3000 km maintained at cheapest cost, this is  $2^n$  choices for  $n$  businesses which is intractable NP-hard problem.

- b. Describe a greedy algorithm that will decide which businesses should be selected. State whether or not your algorithm would return the optimal selection of businesses for the council. 4 marks

You could order ascending cost per km for each business as a best-first search heuristic and select the top  $k$  businesses until 3000 km is reached or just over. The greedy strategy will not be optimal if total exceeds 3000 km, but will be optimal if total exactly 3000 km.

**Question 10** (18 marks)

A charity provides scholarships for students to study at university. It grants hundreds of scholarships to students around Australia every year. Scholarships are awarded to students who:

- have strong academic merit
- come from low socio-economic backgrounds
- live in rural areas.

Scholarship recipients were previously selected by a panel of judges. The charity now wishes to automate the selection process using an artificial intelligence (AI) system trained on scholarship data from the past 20 years.

The charity's panel of judges would like to understand more about AI and how it might be applied to the scholarship selection process.

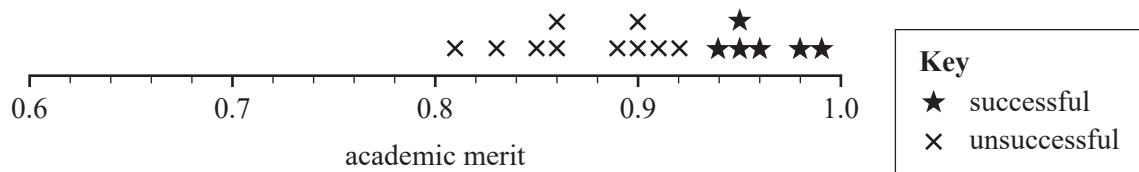
- a. i. Provide a description of Searle's conceptions of strong AI and weak AI for the charity's panel of judges. 2 marks

Searle believes there is no strong AI - strong AI is when an algorithm shows cognition and understanding equivalent to a human. Weak AI in contrast cannot understand and is limited to automated decision making for narrow tasks based on pattern recognition.

- ii. Would a support vector machine (SVM) that selects scholarship recipients be considered strong AI or weak AI? Justify your response. 1 mark

SVM is considered weak AI since it has limited focus (ie scholarship selection) based on a maximum margin classifier identified by the metrics of historic training data.

- b. Academic merit is scored on a scale between 0 and 1. The graph below shows data from a sample of 16 scholarship applications. Academic merit values are shown by the scale, and the success of each scholarship application is indicated by a symbol, as shown in the key.



Describe for the panel of judges what an SVM is and how it could be applied to the classification problem of determining whether a student is awarded a scholarship based on their academic merit.

3 marks

An SVM is a maximum margin classifier algorithm that uses historic data for training weights of feature metrics in order to optimise a hyperplane classifier with the largest margin identified by the support vectors - which are the nearest points of the opposing classes to the hyperplane border.

DO NOT WRITE IN THIS AREA

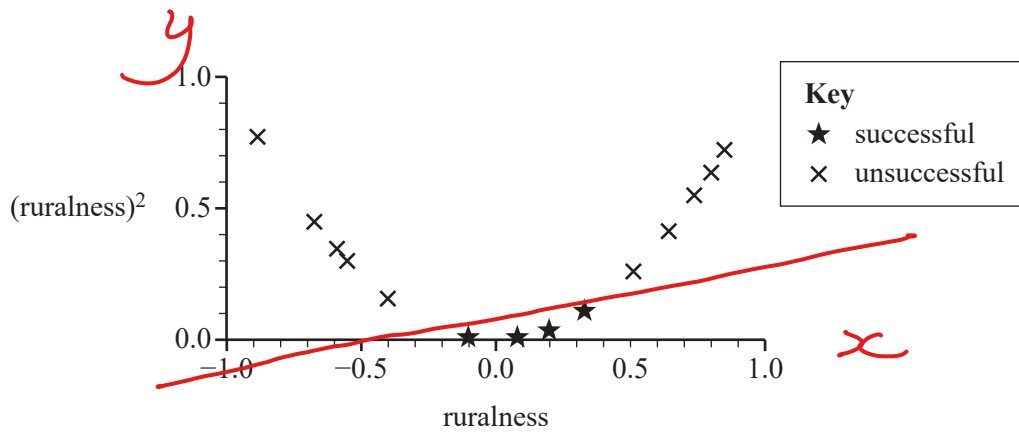
- c. Students may live in major cities, rural areas or remote areas. While it is harder for both rural and remote students to access university education, **the charity focuses on students in rural areas only**, because students in remote areas have other supports available to them. Each student is scored on their ruralness (whether they live in a rural area) on a scale between  $-1$  and  $1$ , with  $-1$  indicating that the student lives in a major city and  $1$  indicating that the student lives in a remote area.

A small sample of ruralness data of past scholarship applicants is shown in the graph below.



If an SVM classifier is going to be trained on this data, explain why it would be useful to create a second dimension from the ruralness data by squaring each value, as shown in the graph below.

2 marks



In order to separate the two classes a kernel transformation is applied to the original 1D data which is not linearly separable. Applying  $x^2 = y$  transform to the 1D data adds a second dimension  $y$  which allows linear separation via an inequality on the cartesian plane - where  $x = \text{ruralness}$  and  $y = (\text{ruralness})^2$ .

DO NOT WRITE IN THIS AREA

- d. The charity has academic merit data and ruralness data from applications for the last 20 years, covering over 20 000 students. A criterion of socio-economic status was only introduced in the last five years, so data for this criterion is only available for about 1000 applications.

It has been observed in the data from the last five years that all scholarship applications were either clearly successful or clearly unsuccessful and that there were no close decisions. In contrast, in prior years there were often instances in which it was difficult to decide who should be awarded a scholarship because applications were similar to each other.

Ethical decision-making is one of the charity's values and it is important to the charity that the process for awarding scholarships is fair, unbiased and transparent.

As an adviser to the charity, consider the options of using an SVM-based system, using a neural-network-based system or of continuing to use a panel of human judges for selecting scholarship recipients. Provide a recommendation to the charity as to which system for awarding scholarships would be most suitable. To do this:

- evaluate the three options that are being considered and the possible choices of training data
- justify your recommendation with consideration of the charity's priorities and the expected quality of the decision-making.

10 marks

SVM creates a maximum margin classifier for identifying a hyperplane for automatic decision making - with confidence proportional to the width of the margin as identified by the support vectors of opposing class (award, not award) if the data is linearly separable in the current or higher dimensions via a kernel transform.

A multilayer neural network can be trained to classify data that is non-linearly separable by identifying a piecewise function classifier - which may be prone to overfitting and may perform poorly with outliers.

Both of these AI machine learning algorithms are limited to the most recent data of 1000 points with the necessary disadvantage metrics.



The human judgement could continue if the number of applications was small as the decisions have been correct for the last 5 years. The historic data lacks information for measuring disadvantage and has resulted in errors, so it should no longer be used. In order to ensure transparency, fairness and explainability if there are too many applications to be handled by humans the SVM algorithm will be a more intuitive and explainable AI solution that can easily be checked by humans for weak classification with identification of outliers or non-outliers with the maximum margin classifier by analysis of any points within the margin.



**Answers to multiple-choice questions**

<b>Question</b>	<b>Answer</b>
1	A
2	D
3	D
4	B
5	C
6	D
7	A
8	A
9	B
10	C
11	A
12	C
13	C
14	C
15	B
16	C
17	A
18	B
19	B
20	D