VICTORIAN CURRICULUM AND ASSESSMENT AUTHORITY

Victorian Certificate of Education 2022

SUPERVISOR TO ATTACH PROCESSING LABEL HERE

STUDENT NUMBER | | | | | | | | | | Letter | |

ALGORITHMICS (HESS)

Written examination

Thursday 10 November 2022

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	13	13	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
$$T(n) = \begin{cases} O(n^c) & \text{if } \log_b a < c \\ O(n^c \log n) & \text{if } \log_b a = c \\ O(n^{\log_b a}) & \text{if } \log_b a > c \end{cases}$$

Question 1

Container ships carry standard-sized shipping containers across the oceans. The containers are loaded in a given order and unloaded in the reverse order at the destination.

Which one of the following abstract data types (ADT) is the most suitable for modelling this situation?

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

Question 2

Which one of the following is the signature for the 'add a new element' operation of an array?

A. element × array → array
B. element → array
C. array → array × element
D. array → element

Question 3

G is a connected, undirected graph with n vertices.

Therefore, G can have

A. between $n - 1$ and $\frac{n(n-1)}{2}$ edges.

B. between $n - 1$ and $\frac{n}{2}$ edges.

C. between n and $\frac{(n-1)}{2}$ edges.

D. between $\frac{n(n-1)}{2}$ and $n(n - 1)$ edges.

Question 4

A year that is divisible by 4 is usually a leap year. The only exceptions to this rule are years that are divisible by 100 but not divisible by 400.

A particular computer program stores year data as a four-digit integer. Let *X*, *Y* and *Z* be three Boolean variables, where:

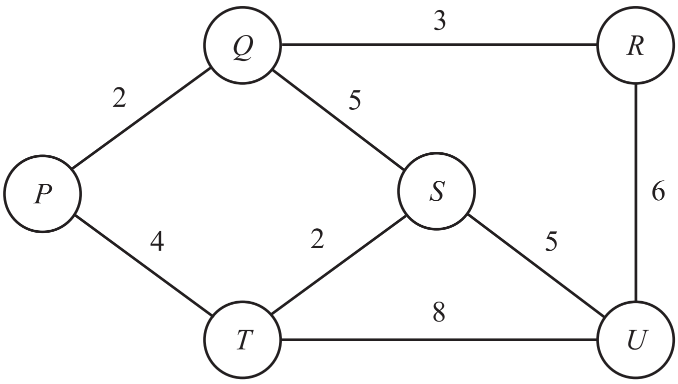
- *X* is 'the year is divisible by 4'
- *Y* is 'the year is divisible by 100'
- *Z* is 'the year is divisible by 400'.

Which one of the following expressions evaluates to TRUE if a year is a leap year?

A. *X* AND *Y* AND *Z*
B. (*X* OR (NOT *Y*)) AND *Z*
C. (*X* AND (NOT *Y*)) OR *Z*
D. *X* AND (NOT *Y*) AND (NOT *Z*)

Use the following information to answer Questions 5 and 6.

The graph *G* is shown below.



Question 5

When Prim's algorithm is run on *G* to find its minimal spanning tree, the order in which the algorithm visits nodes could be

A. *P*, *Q*, *R*, *S*, *T*, *U*
B. *R*, *Q*, *P*, *S*, *T*, *U*
C. *S*, *T*, *P*, *U*, *Q*, *R*
D. *T*, *S*, *P*, *Q*, *R*, *U*

Question 6

Which one of the following is a shortest path from *P* to *U*?

A. *P*, *T*, *U*
B. *P*, *Q*, *S*, *U*
C. *P*, *Q*, *R*, *U*
D. *P*, *Q*, *S*, *T*, *U*

Question 7

Consider the following function.

```
Algorithm foo(n):
    a ← 0
    b ← 0
    While (a < n) Do
        a ← a + 3
        b ← a - b
    End
    Return a – b
```

The value returned by `foo(22)` is

A. 9

B. 12

C. 15

D. 24

Question 8

Which one of the following algorithms would be suitable for determining which node in a graph is the most important?

A. the Bellman-Ford algorithm

B. Dijkstra's algorithm

C. the Floyd-Warshall algorithm

D. the PageRank algorithm

Question 9

The objective of the travelling salesman problem is to find the

A. lowest-cost tree that connects each node in the graph.

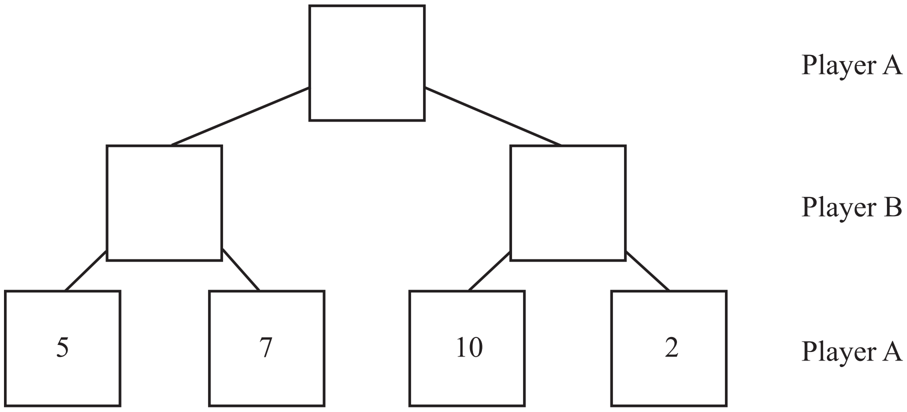
B. lowest-cost paths that connect each pair of nodes in the graph.

C. lowest-cost cycle that traverses every node of the graph exactly once.

D. lowest-cost cycle that traverses every edge of the graph exactly once.

Question 10

The diagram below shows part of the search tree created by the minimax algorithm for a particular problem. Player A is the maximising player and Player B is the minimising player.



The score that the algorithm would assign to the top node in the diagram is

A. 2

B. 5

C. 7

D. 10

Question 11

Which one of the following statements describes the divide step of the quicksort algorithm?

A. Divide the input list at its middle element to create two sub-lists.

B. Merge the two sorted sub-lists with the lower sub-list first.

C. Select an element from the input list. Create two sub-lists by dividing the input list into those values lower than or equal to the element and those values higher than the element.

D. Divide the input list into a sorted sub-list and an unsorted sub-list.

Use the following information to answer Questions 12 and 13.

Leah implements the following algorithm for calculating the nth Fibonacci number.

```
Algorithm fibonacci(n):
    If n = 0 Do
        Return 1
    If n = 1 Do
        Return 1
    prev1 ← fibonacci(n - 1)
    prev2 ← fibonacci(n - 2)
    Return prev1 + prev2
```

Question 12

Let $T(n)$ be a function for the running time of `fibonacci(n)`. For $n \leq 1$, $T(n) = O(1)$.

Which one of the following rules can be used to evaluate $T(n)$ for $n > 1$?

A. $T(n) = T(n-1) + O(1)$

B. $T(n) = 2T(n-1) + O(n)$

C. $T(n) = T(n-1) + T(n-2) + O(1)$

D. $T(n) = T(n-1) + T(n-2) + O(n)$

Question 13

Leah finds that this algorithm runs very slowly for $n > 100$.

It could be made to run significantly faster by

A. storing the results of recursive calls so that they can be re-used.

B. expanding the base case to include `fibonacci(2)= 2`, `fibonacci(3)= 3` and `fibonacci(4)= 5`.

C. combining the two base cases into a single **`If`** statement.

D. rewriting the last three lines of the algorithm as **`Return`** `fibonacci(n - 1) + fibonacci(n - 2)`.

Question 14

Algorithm A is known to be $O(n^3)$.

Which one of the following claims about Algorithm A could **not** be true?

A. Algorithm A is $O(n^2)$.

B. Algorithm A is $\Omega(n^2)$.

C. Algorithm A is $O(n^4)$.

D. Algorithm A is $\Omega(n^4)$.

Question 15

Consider the following recurrence relation.

$$T(n) = \begin{cases} 2T\left(\frac{n}{2}\right) + n & \text{if } n > 1 \\ 3 & \text{if } n = 1 \end{cases}$$

The solution for $T(n)$ using the Master Theorem is

A. $O(n)$

B. $O(n \log n)$

C. $O(n^2)$

D. $O(n^2 \log n)$

Question 16

The Halting Problem was used to demonstrate that

A. we can never know whether a specific computer program will halt or not for a given input.

B. there exist some problems that cannot be solved by an algorithm.

C. we cannot know why a computer program has unexpectedly halted.

D. there exist some true mathematical statements that cannot be proved.

Question 17

Which one of the following methods can demonstrate that a computer can solve all computable problems?

A. Show that the computer can emulate a Turing machine.

B. Show that a Turing machine can emulate the computer.

C. Show that the computer can solve problems in the class NP-complete.

D. Show that the computer can solve problems in both the P and NP-complete classes.

Question 18

Which one of the following describes the method of a greedy heuristic algorithm?

A. Select a local optimum in each step in the hope of progressing towards a global optimum solution.

B. The problem is divided into smaller sub-problems and their solutions are combined to form the solution to the original problem.

C. Generate candidates from the solution space in the hope of finding the global optimum solution.

D. Randomly select one option from several candidates in each step, hoping to eventually obtain a global optimum solution.

Question 19

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

A. NP-complete problems are the hardest problems in NP.

B. All problems in P are also in NP.

C. A proposed solution to a problem in NP can be checked in polynomial time.

D. All problems in NP are NP-complete.

Question 20

Which one of the following statements is **not** true?

A. A complete graph can be acyclic.

B. If $n \geq 3$, a graph with n vertices and at least n edges must be cyclic.

C. If $n \geq 3$, a graph with n vertices must have at least n edges to be cyclic.

D. If an edge is added to a tree, the resulting graph will be cyclic.

END OF SECTION A

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.

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

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

Question 1 (2 marks)

Describe the process of the merge step of the mergesort algorithm when sorting in ascending order.

Question 2 (10 marks)

a. Explain the concept of a decision tree. 2 marks

b. The table below identifies species of birds in terms of their main colours and typical size.

Bird	Colour	Size
mudlark	black and white	20 cm
rosella	red	25 cm
wattlebird	brown	35 cm
magpie	black and white	35 cm
noisy miner	grey	25 cm

In the space provided below, draw a decision tree for identifying the birds shown in the table above using colour as the first decision attribute. 4 marks

c. Let T be a decision tree for identifying birds and let x be a dictionary containing the information about a particular bird. For each non-leaf node u in T, let $P(u, x)$ be a function that returns True if the decision at node u is true for a bird with parameters x and False otherwise.

Write pseudocode for an algorithm that traverses the decision tree to identify a bird based on its features. 4 marks

DO NOT WRITE IN THIS AREA

Question 3 (8 marks)

Hamed and Kashvi are working on a research project to investigate the effect of including separate bicycle lanes in selected streets of a Melbourne suburb. Including a bicycle lane in a street reduces the space for cars. As a result, the cars will have to slow down. This will increase the time it takes drivers to get where they need to go.

Hamed and Kashvi are going to create a simulation of the road network to investigate the effect of introducing bicycle lanes on each road user. First, they will use a data structure to create several road network models for different bicycle lane options. Then, they will use car trip data with each road network model to understand how road users are affected by the different bicycle lane options.

a. What data structure could be used to represent the road network of the suburb for the simulation, including the speeds that are possible on each road? Explain how the features of the road network would be mapped to the data structure. 2 marks

b. Hamed and Kashvi have collected records of typical car trips that people make in the suburb. They have recorded the number of car trips going from each possible origin location to each possible destination.

i. Hamed proposes storing the car trip data using a dictionary in which the key for each entry is the driver and the value stores the origin and destination of the trip. For example:

```
{"Dave": (A,G),
 "Mary": (B,G),
 "Javi": (D,B),
  …
}
```

Kashvi proposes storing the car trip data using a dictionary in which the key for each entry is a particular origin and destination combination and the value stores the number of drivers for that trip on the road network. For example:

```
{(A,B): 3,
 (A,C): 4,
 (A,D): 0,
  …
}
```

The car trip data is going to be used to determine the number of people who are affected by delays on certain roads.

Which of these data representations – Hamed's or Kashvi's – would be more suitable for the simulation? Justify your answer. 2 marks

ii. Let *Trips* be the data representation selected in **part b.i.**

Write signature specifications for the following two operations of *Trips*. 2 marks

- Add a record of a new car trip

add: ______

- Get the total number of car trips

totalTrips: ______

c. Many drivers wish to use the fastest route to their destination and will select their route based on the road speeds at the time they depart.

What algorithm would be suitable for these drivers to use to determine their route before they set off? Justify your answer. 2 marks

Question 4 (11 marks)

A bag contains an unknown number of red and green balls. While there are at least two balls in the bag, the following process is repeated:

- Two balls are picked from the bag at random.
- If both balls are the same colour, a red ball is put into the bag. Otherwise, a green ball is put into the bag.

a. Let $P(n)$ be the statement, 'A bag of n balls will have exactly one ball after the ball-picking process'.

Show by induction that $P(n)$ is true for all integers $n \geq 1$. 3 marks

b. Let $LB(r, g)$ be a function that determines the colour of the last ball in the bag, where r is the initial number of red balls, g is the initial number of green balls, $r \geq 0$, $g \geq 0$, and r and g are not both zero. The function outputs either 'Red' or 'Green'.

i. What do $LB(1, 0)$ and $LB(0, 1)$ evaluate to? 1 mark

ii. Explain why $LB(r, g) = LB(r + 1, g - 2)$ after two green balls are picked. 1 mark

iii. Let *redInPick*(r, g) be a function that simulates a random pick of two balls from the bag and returns the number of red balls obtained.

Write pseudocode for a recursive algorithm for *LB*(r, g). 6 marks

DO NOT WRITE IN THIS AREA

Question 5 (4 marks)

a. Describe how DNA computing overcomes the limitations of traditional sequential search algorithms. 2 marks

b. Describe **two** limitations of DNA computing that have slowed its application to real-world problems. 2 marks

DO NOT WRITE IN THIS AREA

Question 6 (5 marks)

a. State the condition that Prim's algorithm uses to select a new edge in each step of the algorithm. 1 mark

b. Prim's algorithm is run twice on a graph *G* and returns different results.

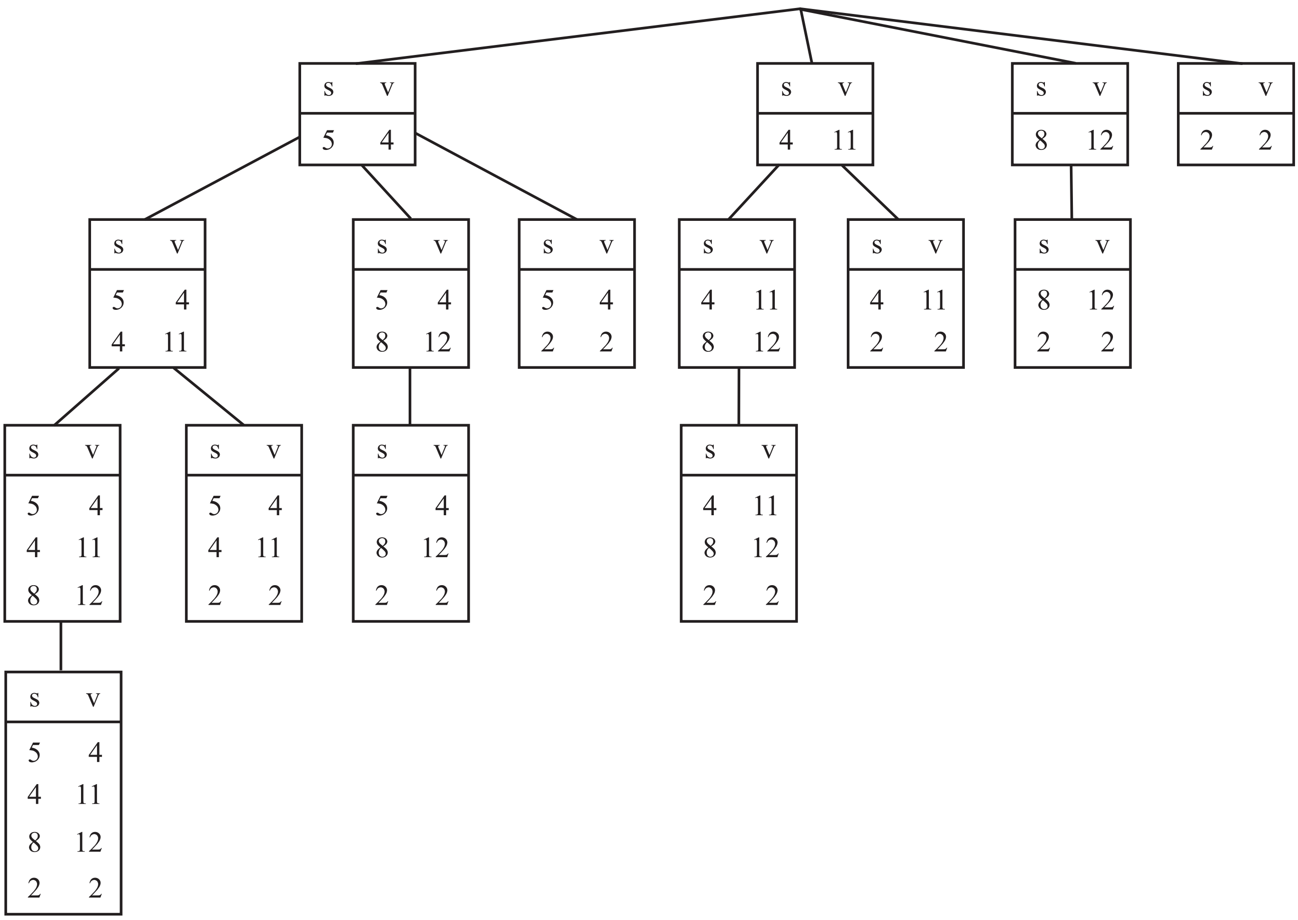
Outline an argument by contradiction for why *G* must contain at least two edges with equal weight. 4 marks

Question 7 (5 marks)

Rosa is solving the knapsack problem with the following items and a knapsack of size 10.

Item	Size(s)	Value(v)
1	5	4
2	4	11
3	8	12
4	2	2

Initially, Rosa produces the search tree below to find the optimal solution.



a. Write pseudocode for an algorithm that uses the following inputs and generates a search tree in the same style as Rosa's:

- a knapsack capacity, *c*
- a list of knapsack item sizes, *sizes*, where `sizes[i]` is the size of item *i*
- a list of knapsack item values, *values*, where `values[i]` is the value of item *i*

4 marks

b. Rosa uses a backtracking algorithm to backtrack whenever she reaches a node that violates the capacity limit of the knapsack.

Circle the nodes in the search tree on page 18 that would **not** be searched by Rosa's algorithm because it backtracks before reaching them. 1 mark

DO NOT WRITE IN THIS AREA

Question 8 (4 marks)

Consider the following algorithm.

```
Algorithm search(Value, List)
   If List is empty Do
     Return False
   split(List, LeftList, MiddleItem, RightList)
   If MiddleItem = Value Do
     Return True
   Else
     LeftResult ← search(Value, LeftList)
     RightResult ← search(Value, RightList)
     Return (LeftResult AND RightResult)
```

`split(List, LeftList, MiddleItem, RightList)` divides `List` and sets the values of `LeftList`, `MiddleItem` and `RightList` such that:

- `LeftList` followed by `MiddleItem` followed by `RightList` is `List`
- if `List` has an odd number of items, then the number of items in `LeftList` and `RightList` are equal
- if `List` has an even number of items, then `LeftList` has one less item than `RightList`.

This is shown in the following two examples:

Example 1

`split([1,2,3,4], LeftList, MiddleItem, RightList)` results in

- `LeftList = [1]`
- `MiddleItem = 2`
- `RightList = [3,4]`

Example 2

`split([1,2,3,4,5], LeftList, MiddleItem, RightList)` results in

- `LeftList = [1,2]`
- `MiddleItem = 3`
- `RightList = [4,5]`

The `split` operation runs in time that is proportional to the size of the `List` input.

Let n be the length of `List`.

a. State the worst case time complexity of this algorithm in terms of n and provide an example of an input `Value` and `List`, with $n \geq 5$, that would result in this worst case. 2 marks

b. State the best case time complexity of this algorithm and outline the relationship between the `Value` and the elements of `List` for this best case to occur. 2 marks

DO NOT WRITE IN THIS AREA

Question 9 (5 marks)

Philippe is implementing the Floyd-Warshall algorithm to find the transitive closure of a graph. He stores the names of the nodes in the graph as a list called `nodes`. For each node in the graph, he stores its edge data using a list of its adjacent nodes. These lists are stored in the dictionary `edges`, where `edges[u]` is a list of the nodes adjacent to node `u`. Philippe plans to implement the algorithm according to the following pseudocode.

```
# checks if nodes u and v are adjacent
Algorithm isAdjacent(nodes, edges, u, v):
  Foreach value in edges[u] as neighbour Do
    If neighbour = v Do
      Return True
  Return False

# implements Floyd-Warshall transitive closure
Algorithm floydWarshall(nodes, edges):
  Foreach value in nodes as i Do
    Foreach value in nodes as j Do
      Foreach value in nodes as k Do
        If isAdjacent(nodes, edges, i, k)
            AND isAdjacent(nodes, edges, j, k) Do
          Append i to edges[j]
          Append j to edges[i]
```

a. Appending an item to one of the edge data lists is a constant time operation. Let *n* be the number of nodes in the graph.

Analyse the time complexity of Philippe's implementation as a function of *n* and hence determine the Big-O time complexity of his `floydWarshall` algorithm. 3 marks

b. Consider graphs with *n* nodes.

Describe a family of graphs that would have a worst case asymptotic running time as a function of *n* for Philippe's `floydWarshall` algorithm. 1 mark

c. How could Philippe's implementation of the Floyd-Warshall algorithm be changed so that it has an improved asymptotic time complexity? 1 mark

Question 10 (2 marks)

Below is a partially complete instruction table for a Turing machine. This Turing machine is initialised to start at the leftmost cell of data and in state A. For an input with an even number of 1s, the output string is the same as the input string. For an input with an odd number of 1s, the output string has a 1 appended to the string. For example, the inputs 0101 and 0111 generate the outputs 0101 and 01111 respectively.

State	**Blank**	**0**	**1**
A	**X**	0, right, A	1, right, B
B	**Y**	0, right, B	1, right, A
halt			

Find the entries in the table for **X** and **Y**.

X ______________________________ **Y** ______________________________

DO NOT WRITE IN THIS AREA

Question 11 (4 marks)

Jia is fascinated by maps and she wants to start a business that allows customers to send her maps to colour and then print. Jia will artistically colour the maps using her favourite three colours of green, gold and purple. No two adjacent regions of a map will have the same colour. She will encourage customers to send maps with any number of regions and any arrangement.

a. Describe two problems with Jia's idea. 2 marks

Problem 1 ______________________________

____

____

Problem 2 ______________________________

____

____

b. Explain how Jia may overcome the two problems described in **part a.** in order to maintain as much of her original idea as possible. 2 marks

Problem 1 ______________________________

____

____

Problem 2 ______________________________

____

____

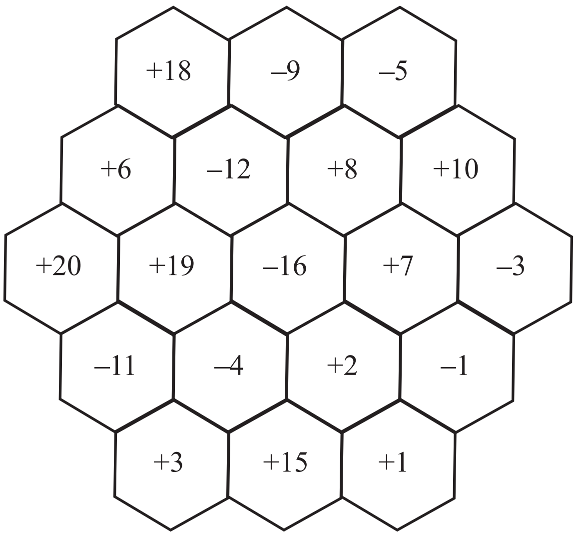
Question 12 (11 marks)

HexaReverso is a single-player game played with hexagonal tiles. Each tile has an integer value, with one side having a positive value and the reverse side having the negative of that value. The tiles are arranged in a hexagonal shape. Large hexagonal grids can be hundreds of tiles wide.

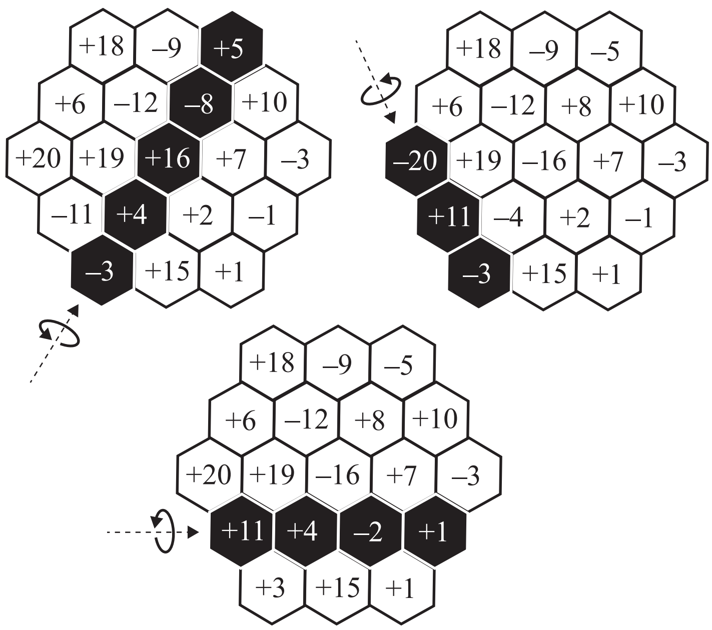
The goal of the game is to maximise the sum of the face-up values on the tiles.

A ‘row’ refers to a straight line of sequentially adjacent tiles. In the game, a player may flip any row of tiles to its reverse side. Three examples of this are shown in the diagram below. The player may flip any number of rows. It is known that this flip operation does not allow for all possible grid arrangements to be generated.

An example of a small HexaReverso board



Three examples of the flip operation



a. It is important that tiles in a particular row can be efficiently identified.

i. Describe how data about the tiles in a HexaReverso game could be stored. A single ADT or a combination of ADTs may be used. 3 marks

ii. Explain how the flip operation would be performed within the data structure described in **part a.i.** 3 marks

b. **i.** What features of this game indicate that a heuristic approach might be needed to achieve the goal of the game? 2 marks

ii. What would be a limitation of using a heuristic approach? 1 mark

c. Kim is going to apply the simulated annealing meta-heuristic to the goal of this game.

i. Describe how she could generate a new candidate solution from a current solution. 1 mark

ii. Kim has correctly implemented the simulated annealing algorithm, but finds that it only accepts candidate solutions that are improvements on the current solution.

Describe how she could modify the parameters of the algorithm to correct this issue. 1 mark

Question 13 (9 marks)

Ishan is discussing what he has been learning in Algorithmics with Yan.

Ishan: 'Hey Yan, today we learnt that it's impossible to prove the correctness of mathematical statements using computers.'

Yan: 'What do you mean? My maths teacher showed me just the other day how to get my CAS calculator to tell me whether an equation is true.'

a. How could Ishan refine his initial statement to Yan to better explain what he has learnt about decidability and refute Yan's counterexample? 3 marks

Ishan continues his discussion with Yan about what he has been learning in Algorithmics.

Yan: 'So, Alan Turing showed that Hilbert's Program was impossible by proving that there are some things that Turing machines cannot do.'

Ishan: 'Yes, Turing showed that Turing machines can only run very simple algorithms, such as those that terminate in polynomial time.'

b. Explain why Ishan's statement is incorrect. Use the connection between Hilbert's Program and Alan Turing's work on Turing machines to support your answer. 4 marks

Ishan and Yan's discussion continues and they talk about artificial intelligence (AI).

Yan: 'Alan Turing must have believed in strong AI as he thought computers could play chess and write poetry. Do you believe in strong AI?'

Ishan: 'Not really. I think Searle's Chinese Room Argument shows that strong AI is impossible.'

c. In the Chinese Room Argument, Searle describes a man in a room who has a book of instructions that allows him to reply to messages he receives that are written in Chinese.

Explain how the Chinese Room Argument is used to argue against the possibility of strong AI. 2 marks

END OF QUESTION AND ANSWER BOOK