VICTORIAN CURRICULUM AND ASSESSMENT AUTHORITY

Victorian Certificate of Education 2020

SUPERVISOR TO ATTACH PROCESSING LABEL HERE

Letter

STUDENT NUMBER

ALGORITHMICS (HESS)

Written examination

Wednesday 25 November 2020

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 24 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

Array
set: element × array × integer → array
lookup: integer × array → element

Dictionary
add: element × dictionary → dictionary
remove: element × dictionary → Boolean

Queue
enqueue: element × queue → queue
dequeue: queue → queue

Stack
push: element × stack → stack
pop: element × stack → element

Which two of the partial abstract data type (ADT) specifications shown above are correct?

A. array and queue
B. array and stack
C. dictionary and queue
D. dictionary and stack

Question 2

Which one of the following terms best describes the process of breaking up a program into small sections that can be easily re-used or updated?

A. reduction
B. modularisation
C. decrease and conquer
D. recursion

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

G is an undirected, connected, acyclic graph with 2020 edges.

What is the minimum number of nodes that *G* could have?

A. 2018

B. 2019

C. 2020

D. 2021

Question 5

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 6

Which one of the following conditional expressions is necessarily **false** regardless of the values of X, Y and Z?

A. (X or not (Y or Z)) or (Y or (X and Z))

B. (X and not (Y or Z)) and (not (X) or (Y and Z))

C. (X or not (Y and Z)) or (not (X) and (Y or Z))

D. (X and not (Y and Z)) and (Y and (X or Z))

DO NOT WRITE IN THIS AREA

Question 7

Consider the following algorithm.

```
Algorithm walk(n, count)
    If count > 0:
        If n is divisible by 3:
            Return walk(n ÷ 3, count – 1)
        Else
            Return walk(2n + 1, count – 1)
    Return n
```

What does the algorithm `walk` return with inputs $n = 10$ and $count = 7$?

A. 7

B. 23

C. 47

D. 95

Question 8

Which one of the following statements describes an advantage of breadth-first search relative to depth-first search on an unweighted graph?

A. Breadth-first search is easier to implement.

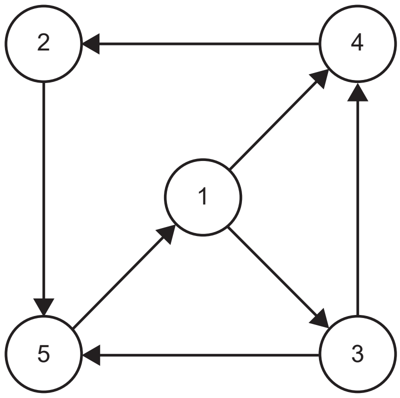
B. Breadth-first search can be modified to calculate the distances of all vertices from the source.

C. Breadth-first search is guaranteed to find the target node since it checks all adjacent vertices.

D. Breadth-first search uses a stack to efficiently check each vertex.

Use the following information to answer Questions 9 and 10.

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 9

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 10

The correct PageRank for Node 3 after two iterations is

A. 0.1150

B. 0.1259

C. 0.1457

D. 0.1639

Question 11

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.

Question 12

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

A. The function $2n^3 + n^2 + 5$ is $O(n^4)$.

B. The function $2n^3 + n^2 + 5$ is $O(n^3)$.

C. The function $2n^3 + n^2 + 5$ is $\Omega(n^4)$.

D. The function $2n^3 + n^2 + 5$ is $\Omega(n^3)$.

Question 13

Consider the recurrence relation

$$T(n) = \begin{cases} 4T\left(\dfrac{n}{2}\right) + n^2 & \text{if} \quad n > 1 \\ 6 & \text{if} \quad n = 1 \end{cases}$$

$T(n)$ specifies a function that is

A. $O(n^2)$

B. $O(n \log n)$

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

D. $O(n^4)$

Question 14

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

A. Quicksort has a worst case time complexity of $O(n \log n)$.

B. Quicksort has a worst case time complexity of $O(n^2)$.

C. Quicksort has a worst case time complexity of $\Omega(n^3)$.

D. Quicksort has a worst case time complexity of $\Theta(n \log n)$.

Question 15

Consider the following algorithm.

```
Algorithm myProcedure(n)
Begin
    If n = 1 Then
        print("*")
    Else
        myProcedure(n – 1)
        myProcedure(n – 1)
    EndIf
End
```

The number of *'s printed by `myProcedure(n)` is

A. O(1)

B. O(n)

C. O(n^2)

D. O(2^n)

Question 16

Which one of the following best describes the process of the mergesort algorithm?

A. Separately sort the left and right halves of the list, and then combine the two sorted lists.

B. Divide the list into two lists formed by those less than or greater than the first element in the list, sort these two lists, and then join the less-than list, first element and greater-than list.

C. Search the list for the lowest element and place it at the front of the list. Repeat this, looking for the next lowest and placing it after the last sorted element, until the entire list is sorted.

D. Consider each element of the list in turn, moving it left in the list past any values greater than it.

Question 17

Which one of the following is **not** a limitation of heuristic algorithms?

A. Some of the solution space may not be considered.

B. The solution returned may not be optimal.

C. They may be too slow to be useful.

D. The solution may not converge on a good result.

Question 18

In a simulated annealing algorithm, which one of the following is affected by the temperature?

A. the score of a new candidate solution

B. the acceptance probability of a new candidate solution

C. the time it takes to generate a new candidate solution

D. the maximum score of a new candidate solution

Question 19

Which one of the following questions **cannot** be decided by a Turing machine?

A. Are the integers u and v equal to each other?

B. Are the real numbers m and n equal to each other?

C. Are the cyclic graphs K and L equal to each other?

D. Are the acyclic graphs G and H equal to each other?

Question 20

Which one of the following is **not** a necessary characteristic of a Turing machine?

A. The head must be able to jump to an arbitrary cell on the tape.

B. The tape must be divided into cells.

C. The table of instructions must be finite.

D. The state register must store the current state of the machine.

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.

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

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 (5 marks)

a. Explain, with an example, the concept of the dictionary abstract data type (ADT). 2 marks

b. Write a complete signature specification for a dictionary ADT. 3 marks

Question 2 (5 marks)

At an airport, incoming aeroplanes need to obtain two types of permission from an air traffic controller in order to land:

1. permission to enter the controlled airspace of the airport, which is granted on a first come, first served basis
2. permission to land, which is granted based on a combination of different factors

a. Explain why a queue ADT is the most appropriate choice for modelling the ordering of aeroplanes seeking permission to enter the controlled airspace of the airport. 1 mark

b. Once the aeroplanes enter the controlled airspace, they need to wait for approval from the air traffic controller to land. This permission is granted based on the following three factors:

- how much fuel remains in the aeroplane's fuel tank
- how far behind schedule the aeroplane is
- whether the aeroplane is local or not (local aeroplanes are given priority)

Describe a combination of ADTs that would allow the air traffic controller to efficiently access the three factors above about each aeroplane in order to determine which aeroplane will be next to land. Justify why each ADT is required and what aspect of the situation each ADT is modelling. Do **not** attempt to define a specific function that calculates which aeroplane will land next. 4 marks

DO NOT WRITE IN THIS AREA

Question 3 (6 marks)

a. Write a definition for a directed graph ADT. 2 marks

b. When modelling real-time information for the road network of a particular city, it is common to use a directed and weighted graph ADT.

i. Explain, using an example, why including directed edges is appropriate for modelling for a road network. 2 marks

ii. Explain, using an example, why including weighted edges is appropriate for modelling for a road network. 2 marks

Question 4 (4 marks)

Consider an unweighted graph G.

a. Write a definition for the transitive closure of G. 2 marks

b. The Floyd-Warshall algorithm for transitive closure could be used to find the transitive closure of G. If the graph is assigned uniform edge weights of one unit, Floyd's algorithm for the all-pair shortest path problem could also be used to find the transitive closure of G.

Outline an alternative approach to finding the transitive closure of G that uses neither of the algorithms above. 2 marks

Question 5 (7 marks)

The mayor of a city would like to model how to dispatch fire trucks to fires in the city. The following information is known:

- There are n fire stations in the city and each fire station in the city has one fire truck.
- On receiving a fire alert, the system will dispatch a fire truck to an intersection of roads. If the fire is not at an intersection, the nearest intersection is automatically nominated as the destination of the truck.
- Traffic conditions in the city are always exactly the same.

a. Describe and justify an appropriate graph ADT that could be used to model the information required to dispatch fire trucks. 2 marks

b. The city wants to pre-calculate the travel time required from each fire station to each intersection in the city. The city is considering using either Dijkstra's algorithm or Floyd's algorithm to solve this problem. The running time of each algorithm is given below.

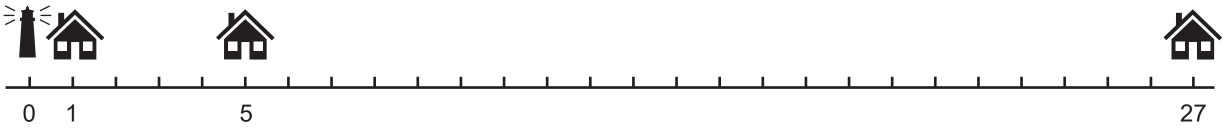
Dijkstra's algorithm for the single-source shortest path problem	$O((\lvert V\rvert + \lvert E\rvert)\log\lvert V\rvert)$
Floyd's algorithm for the all-pair shortest path problem	$O(\lvert V\rvert^3)$

By considering lower and upper bounds of $|E|$ (the number of edges in a graph), compare the running times of the algorithms as applied to the problem of pre-calculating the travel time from each fire station to each intersection in the city. 5 marks

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.

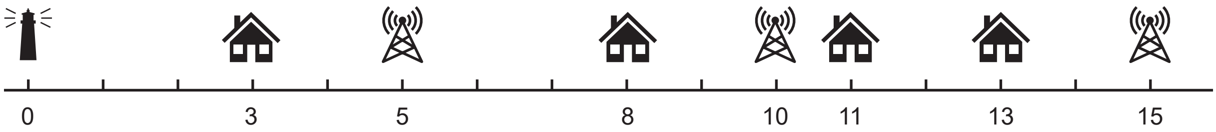
a. Describe a brute-force algorithm that would solve the problem of finding the smallest number of towers required to provide reception to all houses. 3 marks

b. Explain whether the brute-force algorithm described in **part a.** is feasible. Do **not** attempt to derive an exact time complexity. 2 marks

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

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

DO NOT WRITE IN THIS AREA

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

---

b.

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

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

---

---

c. Calculate the time complexity of the club's algorithm. 2 marks

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

DO NOT WRITE IN THIS AREA

Question 8 (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
⋮	⋮	⋮

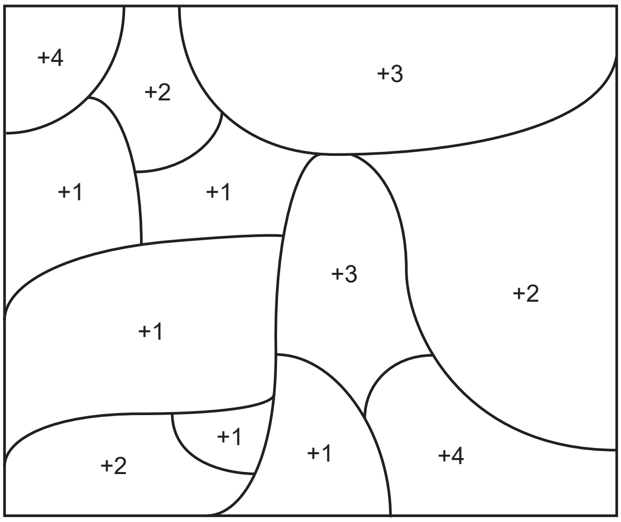
The Chief Operating Officer for the council needs to determine which businesses should be awarded the work so that the cost for the council is minimised.

a. Explain whether or not the problem of finding the optimal collection of businesses to maintain the roads is tractable. 3 marks

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

Question 9 (12 marks)

TerraQuesta is a game played on a board that is divided into regions and each region has a score value. There are many different game boards, and these boards can vary in both size and layout. Larger boards have hundreds of regions. An example of a small board is shown below.



When playing the game, players select regions of the board to own until no further selection is possible. A player cannot select a region that is adjacent to a previously selected region. The game has a single-player version and a two-player version.

Nick, a keen player of TerraQuesta, is trying to beat his previous high score. In the single-player version of the game, the aim of the game is to select regions so that the total score is maximised.

a. Describe a randomised heuristic algorithm that Nick could use to find a set of regions to beat his previous high score. 4 marks

Instead of writing a randomised heuristic algorithm, Nick develops the following algorithm for the single-player version of the game. The algorithm numbers each region of the board and constructs solutions as arrays. In solution arrays, if `solution[v] = 1` then region `v` has been selected. For all regions where `solution[v] = 1`, the solution is valid only if none of these regions are adjacent on the game board.

```
Initialise best_score to −∞
Initialise solution_stack to an empty stack
Push an empty array onto solution stack
While solution_stack is not empty
    Pop the top item off solution_stack as current_solution
    Check the validity of current_solution
    If length of current_solution is equal to the number of regions
        Compare its score to best_score and if it is an improvement then
        save current_solution and its score as best_score and
        best_solution, respectively
    Else
        Push two extensions of the current solution onto solution_stack,
        one with a 0 appended and one with a 1 appended
Return best_solution
```

b. Name the algorithm design pattern used in this algorithm. 1 mark

___

c. What is the worst case time complexity of Nick's algorithm? Explain how you determined this time complexity and its implications for solving game boards with hundreds of regions. 3 marks

___

___

___

___

___

___

d. In the two-player version of the game, each player takes turns selecting a region and the winner is the player with the highest total score.

Explain how the minimax algorithm could be applied to this game. Identify how a score could be determined for each node of the minimax search tree and how accommodations could be made to limit the time the algorithm runs for. 4 marks

Question 10 (6 marks)

In his 1927 program to fully formalise mathematical reasoning, David Hilbert outlined the following three goals:

- Mathematics should be **complete**.
- Mathematics should be **consistent**.
- Mathematics should be **decidable**.

a. Define what is meant by any two of the goals above. 2 marks

1. ______________________________

2. ______________________________

b. Describe the Halting Problem. 2 marks

c. Explain why Alan Turing's formulation of the Halting Problem made Hilbert's program impossible. Include a reference to the specific goal in Hilbert's program that is contradicted by the Halting Problem. 2 marks

Question 11 (3 marks)

Amira and Ming are each trying to solve one of two different decision problems.

Amira says that, for her problem, the best algorithm she can find terminates with the correct answer on all inputs except 5, on which it does not terminate, and that for an input of size n, the algorithm takes at most $n^2 + n \log n$ time.

Ming says that, for her problem, she has found an algorithm that terminates with the correct answer for all inputs, but for an input of size n her algorithm takes somewhere between $n!$ and n^n time.

Based on this information, what can be deduced about the decidability of the problems that Amira and Ming are working on? Justify your answer.

Question 12 (4 marks)

The three-colouring problem asks if a graph can be coloured with three colours.

Peter is investigating the three-colouring problem. He uses an algorithm that takes as input a graph with n vertices and an array of size n specifying one of three colours for each vertex in the graph. The algorithm determines whether or not the given array of colours is a valid three-colouring of the graph and runs in at most $2^n + n^4$ time for an input of size n.

Jane informs Peter that three-colouring is a known NP-complete problem.

a. Explain why Peter's algorithm cannot be used to justify that the three-colouring problem is in NP. 2 marks

b. Jane is investigating a different NP-complete problem – the travelling salesman problem. Jane claims to have found an algorithm that will find a solution to the travelling salesman problem for a graph of n vertices in at most time $n^3 + n^2$.

If Jane is correct, what are the consequences for solving all other NP-complete problems? Explain your answer. 2 marks

Question 13 (3 marks)

Consider an implementation of Dijkstra's algorithm that maintains a set of vertices called Q, whose shortest distances have been finalised, and an array of distances *dist*, where $dist[u]$ is the current shortest known distance to u.

Now, consider the step in Dijsktra's algorithm where node v is added to Q via the edge (u, v) where $u \in Q$.

Outline an argument by contradiction to show that there cannot be a shorter path to node v.

END OF QUESTION AND ANSWER BOOK

DO NOT WRITE IN THIS AREA