Name:

ALGORITHMICS UNIT 3 & 4

Trial Exam 2: 2022

Reading Time: 15 minutes
Writing time: 120 minutes (2 hours)

QUESTION AND ANSWER BOOK

Section	*Number of questions*	*Number of questions to be answered*	*Number of marks*
A	20	20	20
B	8	8	80

- 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 ?? 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 you name in the space provided to verify this.

- All written responses must be in English, point form is preferred.

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

The VCAA Exam will include the Master Theorem in this form.

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

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

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

<table>
<tr>
<td>
The VCAA form of Master Theorem is equivalent to the form of Master Theorem taught in our class by consideration of log laws.
$$\log_b a = c \iff a = b^c \iff \frac{a}{b^c} = 1$$
$$\log_b a < c \iff a < b^c \iff \frac{a}{b^c} < 1$$
$$\log_b a > c \iff a > b^c \iff \frac{a}{b^c} > 1$$
</td>
<td>
$$T(n) = aT\left(\frac{n}{b}\right) + f(n^k)$$

$\frac{a}{b^k} < 1$ then $O(n^k)$
$\frac{a}{b^k} = 1$ then $O(n^k \log_b n)$
$\frac{a}{b^k} > 1$ then $O(n^{\log_b a})$

</td>
</tr>
</table>

SECTION A – Multiple Choice – circle one option only

Question 1

How many times is the while loop executed in the pseudocode below?

```
Algorithm Aces(input X, output Y)
// Aces is a really cool algorithm
// It accepts input X and calculates the  output Y
      While (X > 0) do
            Y:=Y+1
            X:=X-1
      End do
End Algorithm
```

A. X times

B. Unknown

C. Y times

D. X-1 times

Question 2

Which ADT would be most appropriate to model the following scenarios respectively:
(i) Medication(s) taken by patient X
(ii) Routes travelled by council garbage truck on different days.

A. (i) Graph, (ii) Array

B. (i) List, (ii) Graph

C. (i) Stack, (ii) Graph

D. (i) List, (ii) Queue

Question 3

Which ADT would be most appropriate to model the following scenarios respectively:
(i)The geographic location of each member country of the United Nations and their trading links.
(ii) Major export items in order of GDP contribution of each United Nation member country

A. (i) Graph, (ii) Priority Queue

B. (i) Dictionary, (ii) List

C. (i) Tree, (ii) List

D. (i) List, (ii) Stack

Question 4

A Hamiltonian path in a graph is defined by:

A. A tour of a graph where each node is visited once only

B. A tour of a graph where each edge is traversed once only

C. A tour of a graph that gives the shortest path from one node to another node

D. A tour of a graph where each node and each edge are visited once.

Question 5

For a graph G=(V,E) the number of minimum edges required for a **connected** graph and the number of edges required for a **complete** graph are respectively:

A. $|V|-1, \quad \frac{|V|(|V|-1)}{2}$

B. $|V|+2|V|, \quad |V|-1$

C. $|V|(|V|-1), \quad |V|$

D. $|V|+2, \quad \frac{|V|(|V|-1)}{2}$

Question 6

Given two vertices in a graph s and t, which of the two traversals (BFS and DFS) can be used to find if there is path from s to t?

A. Only BFS

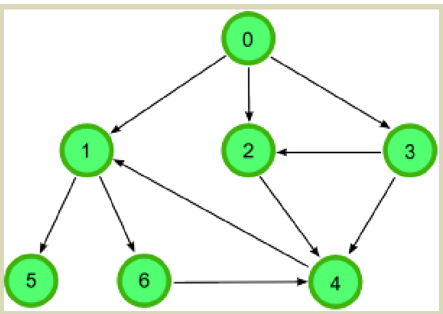
B. Only DFS

C. Both BFS and DFS

D. Neither BFS nor DFS

Question 7

Consider the following graph



A possible sequence of nodes visited using Depth first search is:

A. 0, 1, 2, 3, 4, 5, 6

B. 0, 2, 4, 1, 3, 5, 6

C. 0, 3, 4, 2, 1, 5, 6

D. 0, 1, 5, 6, 4, 2, 3

Question 8

A Brute force algorithm is one that:

A. Systematically explores all possible options before determining the optimal solution.

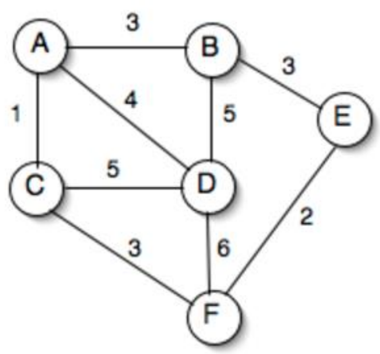
B. Selects the most optimal value at each decision point

C. Randomly generates several solutions and picks the best one

D. Explores most options and uses a probability function to determine the best solution

Question 9

When Prim's is executed on the following graph, starting at any vertex:

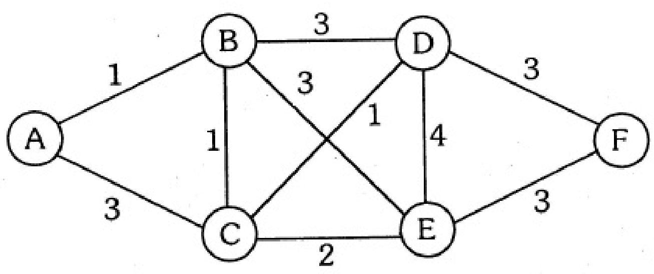


The Edges (in no particular order) included and the weight of the MST are:

A. E={AC,AB,BE,DF,AD} weight=13
B. E={AC,BD,CF,EF,BE} weight=13
C. E={AC,AB,BE,EF,BD} weight=14
D. Both E={AC,AB,BE,EF,AD} weight=13 and E={AC,AD,CF, EF,BE} weight=13

Question 10

Dijkstra's algorithm on the following graph from node A , explores the nodes in order:



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

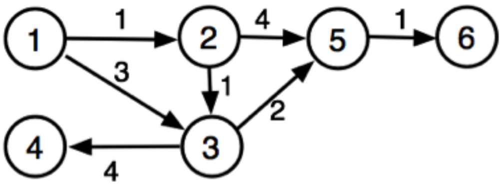
B. A,C,B,E,D,F

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

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

Question 11

Given the following weighted graph:



Which node is **not expanded** by Dijkstra’s algorithm in computing shortest path from Node 1 to Node 5?

A. 2

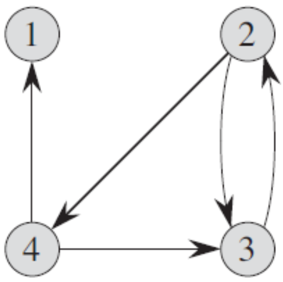
B. 3

C. 5

D. 6

Question 12

Consider the following Directed Graph, the Floyd-Warshall Transitive closure algorithm produces the which matrix:



A

to nodes

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

from

C

to nodes

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

from

B

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

D

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

Question 13

What is the result of the following recursive algorithm, when called with the ***hello(32)*** ?

```
Function hello(input value)
   If (x < 2) then
       Report value
   Else
       Report value +  hello(0.5*value)
   End if
End function
```

A. 32+16+8+4+2+1

B. 32+32+32+32+32+32

C. 32+18+9+5+3+1

D. 1+2+4+6+16+32

Question 14

Consider the following Data Signature on the Priority Queue ADT, which statement is true.

$$
\begin{aligned}
&\textbf{\textit{name prioQueue}};\\
&\textbf{\textit{import element, boolean}};\\
&\textbf{\textit{operations}} \quad \textbf{\textit{newPrioQueue}} : \rightarrow \textbf{\textit{prioQueue}};\\
&\qquad \textbf{\textit{enqueue}} : \textbf{\textit{prioQueue}} \times \textbf{\textit{element}} \times \textbf{\textit{integer}} \rightarrow \textbf{\textit{prioQueue}};\\
&\qquad \textbf{\textit{minElement}} : \textbf{\textit{prioQueue}} \rightarrow \textbf{\textit{element}};\\
&\qquad \textbf{\textit{dequeueMin}} : \textbf{\textit{prioQueue}} \rightarrow \textbf{\textit{prioQueue}};\\
&\qquad \textbf{\textit{isEmpty}} : \textbf{\textit{prioQueue}} \rightarrow \textbf{\textit{boolean}};
\end{aligned}
$$

A. The input parameters for the operation *enqueue* are *PrioQueue, element, integer*.

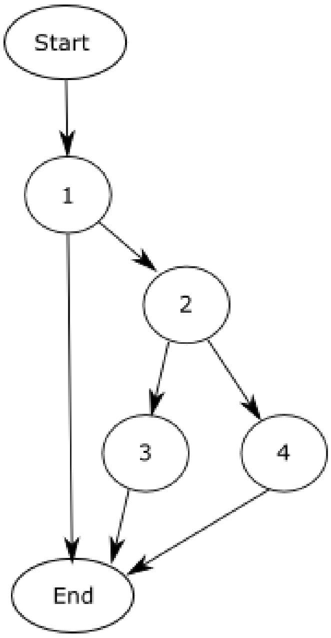
B. The input parameters for the operation is*Empty* is *prioQueue* and *Boolean*.

C. The output from the operation *minElement* is *prioQueue* and *elemeni*.

D. The output from the operation *enqueue* is *element* and *prioQueue*.

Question 15

Which paths need to be tested for a brute force coverage on the following control flow graph?



A. {Start,1,2,3,End} and {Start,1,2,4,End}

B. {Start,1,2,3,End} and {Start,1,2,4,End} and {Start,1,End}

C. {Start,1,2,3,4,End} and {Start,1,End}

D. {Start,1,End} and {Start,1,2,3,End} and {Start,2,4,End}

Question 16

If an argument is another name for the inputs passed into a function then which of these statements is true about the following code?

```
function mystery(input:integer n)
    if (n>0) then
        return n + mystery(n-1)
    else
        return 0
    end if
end function
```

A. The base case for this recursive method is an argument with any value which is less than or equal to zero.
B. The base case for this recursive method is an argument with any value which is greater than zero
C. The base case for this recursive function is an argument with the value zero.
D. There is no base case

Question 17

Fill in the code to complete the following function for checking whether a word is a palindrome, that is it reads the same forwards as backwards. Example of palindromes: wow, noon, level, racecar, .

```
Function IsPalindrome(Inputs: Array word, integer low, integer high)
  if (high <= low) // Base case
      return true
  else if (character at low position of word ≠ character at high position of word)
      return false
  else
      return ________________________
  end if
End function
```

A. IsPalindrome(word)
B. IsPalindrome(word,low,high)
C. IsPalindrome(word,low,high-1)
D. Is Palindrome(word, low+1, high-1)

Question 18

Consider the following recursive ***Algorithm ABC*** shown below. What is the recurrence relation describing the time complexity as a function of the input size "n"?

```
Algorithm ABC(Input Array, size n)
// Input n size of array to process
// the k-th element of Array is Array[k]
      If (n <=1) then
                 base case reached
      else
            for j = 1 to n – 1 do
                  if Array[j] > Array[j + 1] then
                        swap( Array[j],Array[j + 1] )  // swap elements
                  End if
            Enddo
            ABC(Array, n-1)
      End if
End Algorithm
```

A. $\boldsymbol{T(n) = T(n-1) + c}$

B. $\boldsymbol{T(n) = 2T(n) + c}$

C. $\boldsymbol{T(n) = 2(T(n/2)) + c}$

D. $\boldsymbol{T(n) = T(n-1) + n}$

Question 19

After k iterations of Bellman-Ford's algorithm in a graph of n nodes $(1 < k < n-1)$, then which following statement is true about the shortest paths from the source node to every other node?

A. The shortest paths with at most k edges from the source node have been found.

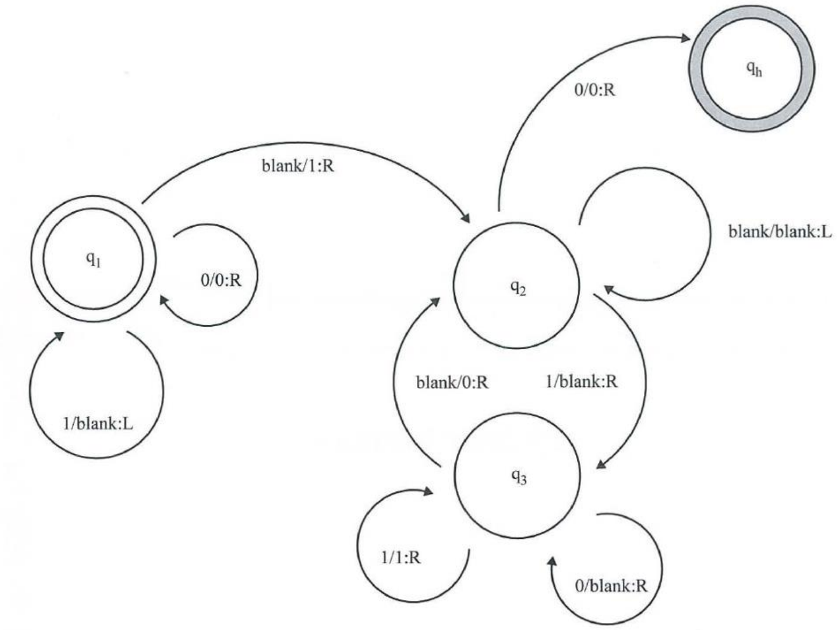
B. The shortest paths with at most $k-1$ edges from the source node have been found.

C. The shortest paths with at most $k+1$ edges from the source node have been found.

D. The shortest paths for any number of edges can change as the algorithm has not completed.

Question 20

A Turing machine is configured with the instructions represented in the state diagram below:



Each edge is labelled ***i/j:k*** where:

- ***i*** is the input symbol
- ***j*** is the output symbol
- ***k*** is the direction the head moves (L=left, R=right) after the output

The machine begins in state q_1 and halts in state q_h.
The machine is given the following tape. For this machine, the tape remains stationary while the head moves.
The arrow shows the starting position of the head.

	0	1	blank	1	1	1	blank	0	H
	↑								

Which one of the following best represents the tape's appearance and the position of the head when the Turing machine halts?

A.

	0	1	blank	1	1	1	blank	0	H
									↑

B.

	0	blank	0	blank	1	1	0	0	H	
										↑

C.

	0	blank	0	blank	1	1	0	0	H
									↑

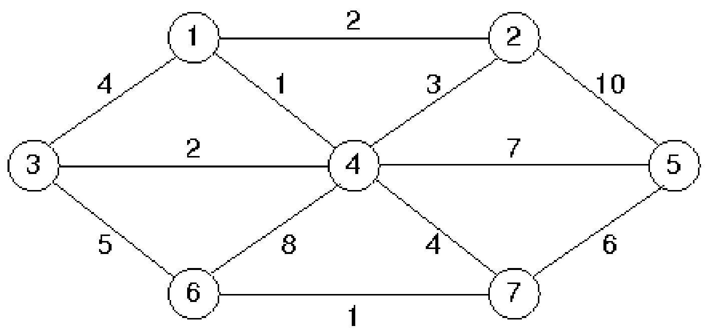
D.

	0	1	blank	1	1	1	blank	0	H	
										↑

SECTION B – Extended Response Questions Answer all questions in the space provided.

Question 1 (8 marks)

a) Why is Prim's MST a greedy algorithm? Describe the characteristics of "Greedy". (2 marks)



b) For the graph shown above, draw the minimum spanning tree using Prim's Algorithm. State the cost of the tree. (2 marks)

c) How many operations are required by Prim's as a function of the number of vertices of a connected graph, discuss the time complexity. (2 marks)

d) Is a minimum spanning tree always unique? Justify your response. (2 marks)

Question 2 (10 marks)

Consider the following algorithm for Prim's to find the MST.

Input: $\boldsymbol{G=(V,E)}$ a weighted graph
Initialize MST: $\mathbf{T=(}\boldsymbol{V}_{\mathbf{MST}} \mathbf{= \{\},}\ \boldsymbol{E}_{\mathbf{MST}} \mathbf{= \{\}\ )}$
Repeat until $\boldsymbol{V}_{\mathbf{MST}} = \boldsymbol{V}$:

Choose minimal weighted edge $\boldsymbol{\{u, v\}}$ where $\boldsymbol{u}$ is in $\boldsymbol{V_{MST}}$ and $\boldsymbol{v}$ is not
Add $\boldsymbol{v}$ to $\boldsymbol{V}_{\mathbf{MST}}$ and $\boldsymbol{\{u, v\}}$ to $\boldsymbol{E}_{\mathbf{MST}}$

Output: $\mathbf{T=(}\boldsymbol{V}_{\mathbf{MST}}\ \mathbf{,}\ \boldsymbol{E}_{\mathbf{MST}}\ \mathbf{)}$ describe a Minimal Spanning Tree (MST)

a) Prove Prim's Algorithm using the Loop Invariant technique, diagrams are permitted. (4 marks)

b) Write an algorithm that takes a graph G and an edge e that is in the MST of G and returns the amount that the weight of e may be increased and still remain in the MST of G. (6 marks)

Question 3 (12 marks)

a) In your own words describe the characteristics that make a problem tractable or intractable. (2 marks)

b) What is the difference between problems in the class “P” and the problems in class “NP”? Give an example of a problem in “P” . (2 marks)

c) What are the characteristics of an NP-complete problem and how does it relate to P, NP and NP-Hard problems? Give an example of an NP-Complete problem. (2 marks)

d) What strategies can be used to find an “acceptable” solution for an NP-Complete problem? (2 marks)

e) Describe in detail the characteristics of an undecidable problem in Computer Science. (2 marks)

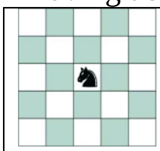
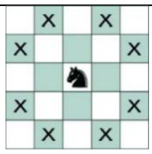
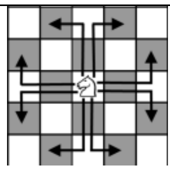
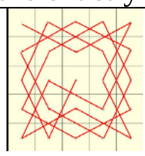
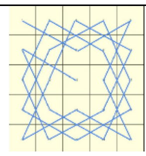
f) Suppose you’re working on a lab for a programming class,, have written your program, and start to run it. After five minutes, it is still going. Does this mean it’s in an infinite loop, or is it just slow and doing calculations? Explain your answers. (2 marks)

Question 4 (10 marks)

a) What properties of problems are best suited for solving using backtracking design patterns? How do Backtracking algorithms compare with naïve Brute Force algorithms in terms of efficiency? (3 marks)

b) Describe how Backtracking algorithms strategies work to solve problems, giving details of the abstract data types (ADTs) that they commonly use. (3 marks)

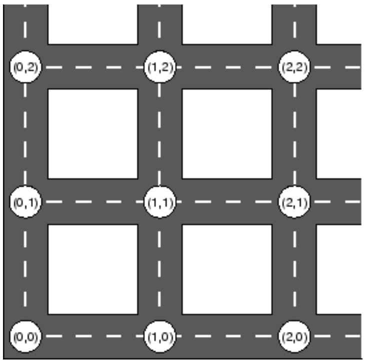
<u>The Knight's Tour problem</u> Given a N*N board with the Knight placed on anywhere on an empty board. Moving according to the rules of chess knight must visit each square exactly once.

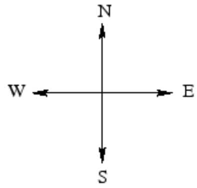
<table>
<tr>
<td>
Place a knight anywhere on a 5x5 chess board to start.</td>
<td>
X's indicate possible moves from start.</td>
<td>
25 cell array</td>
<td>
A possible solution is shown.</td>
<td>
Another possible solution .</td>
</tr>
<tr>
<td colspan="5">Constraint: Knights chess pieces move in an "L-shape" that is: two squares in any direction vertically followed by one square horizontally, or two squares in any direction horizontally followed by one square vertically.</td>
</tr>
</table>

c) Explain the main actions of a Backtracking algorithm using ADTs described in part b) which solves the Knight's Tour problem for an N*N chess board, from any starting position. (4 marks)

Question 5 (10 marks)

Problem: Survival





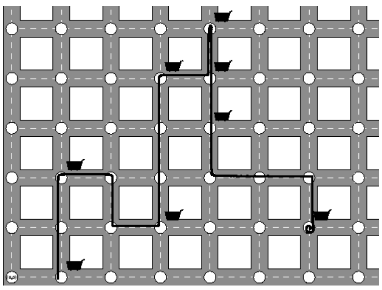
Natural disasters have squashed Coda City and it is now a wasteland roamed by dangerous rats. You have taken refuge in the sewers underneath the city. Fortunately you have your solar powered laptop. Unfortunately, you have no food with you, and so you must leave the sewer to stock up on supplies.

Coda City consists of streets running north-south and east-west, forming a grid. Every street intersection has a manhole. The intersections are given coordinates as shown on the diagram:

Some street intersections contain abandoned shopping trolleys containing food items, cast aside by panicking civilians during the disaster. Since being outside at all is a massive risk, you decide to visit all the shopping trolleys in one trip.

Your plan is as follows: You will emerge from any manhole. You will then run along the streets, emptying trolleys as you go. **You can only travel in an easterly direction, do not head west, as sun rays will blind you.** Once you have been to all the trolleys, you will climb down any manhole.

Your task is, given the locations of abandoned trolleys within the city, to determine the **smallest possible distance** that you must travel above ground in order to collect food from all the abandoned trolleys.

Survival Input	Survival Output
Integer T the number of trolleys List Tcoords a list of (x,y) coordinates showing the position of the k^{th} trolley	should be a single integer which is the shortest distance you must travel above ground to collect all the food from all the trolleys. Remember, you *cannot go west* .
Sample Input	**Sample Output**
Algorithm Survival(8, {(1,0),(4, 3),(3, 4),(4,4),(1, 2),(3,1),(4,5),(6,1)})  The trolleys in the sample data are shown in the diagram above.	16 Explanation The thick black line shows one possible shortest path which visits all trolleys, starting from the bottom-left trolley and ending at the bottom-right trolley. The length of this path is 16. Although many other paths are possible, there are no shorter paths, therefore the answer is 16.

a. If a brute force naïve approach is taken to write an algorithm to find the solution, what will be the worst case time complexity of the naïve approach? Explain your response. (2 marks)

Question 5 (Continued)

b. Write your structured pseudocode algorithm for the Survival problem, using clear and meaningful variable names and using comments to explain any complicated calculations in your algorithm.

(6 marks)

c. Discuss if it is possible for a perfect solution to be found everytime with your algorithm? Explain why.

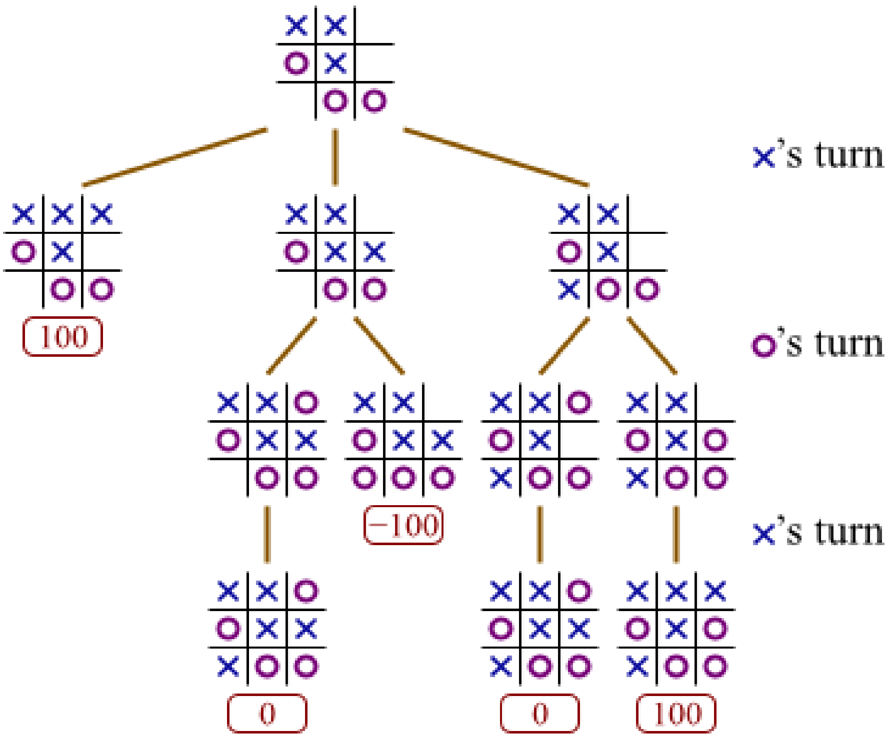
(2 marks)

Question 6 (10 marks)

~~a. Describe in your own words given the pseudocode how the Minimax algorithm works on a game tree.~~
~~(2 marks)~~

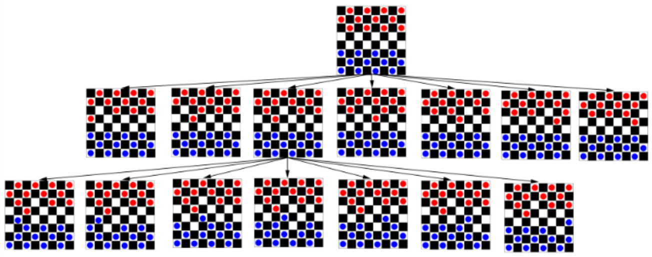
~~In English~~	~~In Pseudocode~~
	~~**function** minimax(node, depth, maximizingPlayer)~~ ~~**if** depth = 0 **or** node is a terminal node~~ ~~**return** the heuristic value of node~~ ~~**if** maximizingPlayer~~ ~~bestValue := -∞~~ ~~**for each** child of node~~ ~~val := minimax(child, depth - 1, FALSE)~~ ~~bestValue := max(bestValue, val)~~ ~~**return** bestValue~~ ~~**else**~~ ~~bestValue := +∞~~ ~~**for each** child of node~~ ~~val := minimax(child, depth - 1, TRUE)~~ ~~bestValue := min(bestValue, val)~~ ~~**return** bestValue~~ ~~**end function**~~ ~~*(* Initial call for maximizing player *)*~~ ~~**minimax(origin, depth, TRUE)**~~

~~b. Label ***all*** internal nodes of the following tic-tac-toe game tree with the value that Minimax algorithm would compute. The leaves have already been labelled.~~ ~~(2 marks)~~



Question 6 - continued

Consider the following subset of game tree for the game of checkers below. In case you didn't know, Checkers is a zero-sum game played by two players on an 8x8 board. Each player begins with 12 counters and the aim is to capture the opponent's counters and remove them from the board.



c. Is it feasible to analyse every possible node in the game tree? Justify your response. (2 marks)

d. Describe how could a heuristic be applied to the checkers game tree? Describe the mathematical principles that could be used to determine a "good guess" in terms of what the next best move might be for a player.

(2 marks)

e. Describe how can you evaluate how good the checkers game heuristic is? Explain your response.

(2 marks)

Question 7 (10 marks)

a) Give the time complexity as a function of n for the following Master Theorems in terms of Big-O notation. (3 marks)

i. $T(n) = 5T\left(\frac{n}{8}\right) + n$

ii. $T(n) = 4T\left(\frac{n}{2}\right) + n^2$

iii. $T(n) = 3T\left(\frac{n}{4}\right) + f(\sqrt{n})$

Function Mergesort (Input List A)	Function Merged(Input List Left, Input List Right)

Function Mergesort (Input List A)
Size=Length(A)
If (Size <=1) **then**
    Return A
else
    List Left = Mergesort([A[1..Size/2])
    List Right = Mergesort([A[Size/2+1..Size])
    Return Merged(Left,Right)
End If
End Function
	Function Merged(Input List Left, Input List Right)
	    While list Left is not empty and list Right is not empty do
	        if (first(Left) <= first (Right)) then
	          append first(Left) to **result** and set Left=rest(Left)
	        else
	          append first(Right) to **result** and set Right=rest(Right)
	        end if
	    End do
	    //If there are still values remaining in either list
	    While not empty Left do
	        append first(Left) to **result** and set Left=rest(Left)
	    End do
	    While not empty Right do
	        append first(Right) to **result** and set Right=rest(Right)
	    End do
	    Return result
	End Function

b) Consider the pseudocode above and without using the Master Theorem ***show or explain why*** the complexity of MergeSort is O(nlogn). Diagrams are permitted. (2 marks)

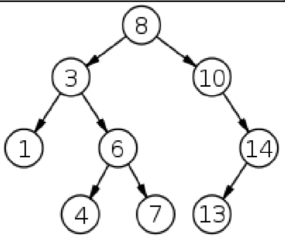
c) Identify all the variables for the Master theorem on the algorithm above and use it to confirm your result from part c) (1 mark)

d) Given an unsorted list of numbers {3,7,9,2} show the detailed steps that Mergesort would use to sort the list. (2 marks)

e) What algorithm design methodology is used in Mergesort to solve the sorting problem? Contrast and discuss how this design pattern compares with Brute Force methods of sorting items. (2 marks)

Question 8 (10 marks)

Consider the following Algorithm Tree2List that converts a Binary Tree into an ordered List.



A binary search tree is a rooted binary tree, whose internal nodes each store a key (and optionally, an associated value) and each have two distinguished sub-trees, commonly denoted *left* and *right*. The tree additionally satisfies the binary search tree property, which states that the key in each node must be greater than all keys stored in the left sub-tree, and smaller than all keys in right sub-tree.

```
Procedure Tree2List(input TreeNode, output NodeList)
    // TreeNode.right follows the right subtree
    // TreeNode.left follows the left subtree
    If (TreeNode has children) then
        Tree2List(TreeNode.right, NodeList)
        Append TreeNode to NodeList
        Tree2List(TreeNode.left, NodeList)
    Else
        Append TreeNode to NodeList
    End if
End procedure
```

a. Label the time complexity of each command on the pseudocode above using the table. (2 marks)

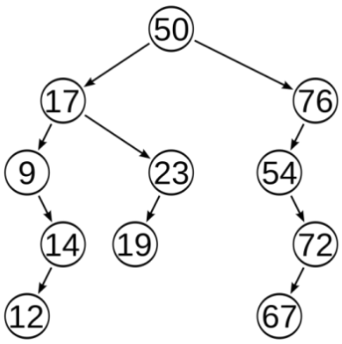
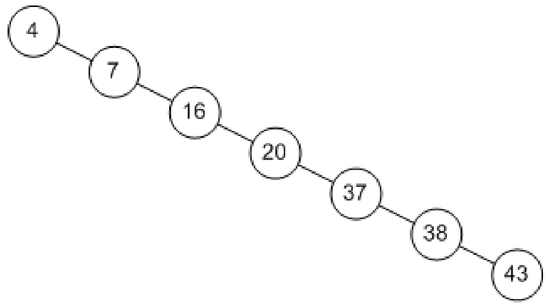
b. Determine the recurrence relation for the time complexity of this algorithm, (2 marks)

c. and hence the time complexity of the algorithm above. (1 mark)

Question 8 (continued)

d. What is the time complexity of <u>searching</u> for a particular value in a binary tree? Explain and justify your answer. (2 marks)

e. Consider the following binary trees and explain how their structure may or may not impact on searching for a particular value. (3 marks)

<table>
<tr><th>Case 1</th><th>Case 2</th></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td></td><td></td></tr>
</table>

END OF TRIAL EXAM 2 (Good Luck!)