NAME:

ALGORITHMICS UNIT 3 & 4

Trial Exam 1: 2019

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	9	9	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 20 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.

SECTION A – Multiple Choice – select one option only

~~**Question 1**~~

~~Which of the following statements is NOT true about DNA programming?~~

A. ~~It provides massive parallel processing power.~~

B. ~~It allows computation using biological molecules and chemical processes.~~

C. ~~DNA "code" can be used to represent symbols and form language.~~

D. ~~It allows NP-Complete problems to be solved.~~

Question 2

What is the result of the following recursive algorithm?

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

When called with the ***hello(32)***:

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 3

The diameter of a graph is:

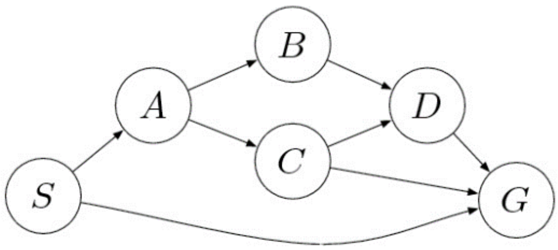
A. The smallest d in a graph, such that every pair of vertices u and v have distance(u,v) ≤ d.

B. The total sum of the edge weights in a graph G.

C. The shortest longest path in a graph G.

D. The number of edges in a Graph.

The following directed graph is to be used to answer Question 4 and Question 5.



Question 4

Using Breadth-First search starting at node S and choosing in alphabetic order when multiple options are available results in the following traversal order:

A. SACBDG

B. SAGCBD

C. SABDCG

D. SAGBCD

Question 5

Using Depth-First search starting at node S and choosing in alphabetic order when multiple options are available results in the following traversal order:

A. SGABDC

B. SABDGC

C. SAGDCB

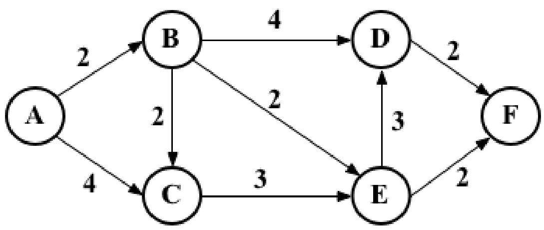
D. SACDGB

Question 6

A decision problem that does not have an algorithm for solving it is classified as a:

A. Intractable problem
B. Undecidable problem
C. Feasible problem
D. Tractable problem

The following weighted, directed graph is to be used to answer Question 7, Question 8 and Question 9.



Question 7

Starting at node A for the following graph, using Dikjstra's algorithm the shortest distances found to the listed nodes is:

A. Node B (distance=2), Node D (distance=6), Node F (distance=8)

B. Node B (distance=2), Node D (distance=6), Node F (distance=7)

C. Node B (distance=2), Node D (distance=6), Node F (distance=6)

D. Node B (distance=2), Node D (distance=7), Node F (distance=6)

Question 8

If every edge of the graph is multiplied by negative one finding the shortest path using Dijkstra's algorithm starting at node A will result in:

A. Dijkstra's algorithm will not work with negative edges

B. Node F (distance= -9)

C. Node F (distance= -6)

D. Node F (distance = - 12)

Question 9

Back to the original graph, if the edge C-E is reversed and multiplied by negative 1, the resulting shortest path from node A to node C is:

A. Dijkstra's algorithm will not work with negative cycles

B. Node C (distance= 1)

C. Dijkstra's algorithm will not work with negative edges

D. Node C (distance= 4)

Question 10

Which of the given options provides the increasing order of asymptotic complexity of functions f1, f2, f3 and f4?

$$f1(n) = 2^n$$
$$f2(n) = n^{3/2}$$
$$f3(n) = n \log n$$
$$f4(n) = n^{\log n}$$

A. *f3*, *f2*, *f4*, *f1*

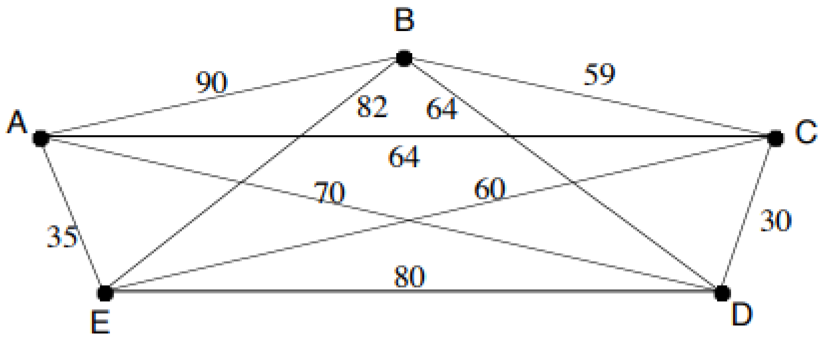
B. *f3*, *f2*, *f1*, *f4*

C. *f2*, *f3*, *f1*, *f4*

D. *f2*, *f3*, *f4*, *f1*

Question 11

Consider the following weighted graph, starting with node A and using the Greedy strategy of nearest neighbour, the Hamiltonian circuit that results visits the nodes in order:



A. A E C D B C

B. A E C D B A

C. A C D B E A

D. A E B D C A

Question 12

Determining whether a Hamiltonian path exists in a graph is a problem that is:

A. NP-Complete

B. NP-Hard

C. Feasible

D. P

Question 13

Consider the following pseudocode. What is the total number of multiplications to be performed?

```
D := 2
for i = 1 to n do
   for j = i to n do
      for k =(j + 1) to n do
           D := D * 3
```

A. Half of the product of the 3 consecutive integers.

B. One-third of the product of the 3 consecutive integers.

C. One-sixth of the product of the 3 consecutive integers.

D. None of the above.

Question 14

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 15

Simulated Annealing is a heuristic used to find acceptable solutions for NP-Complete and NP-Hard problems, it uses the method of:

A. Thinking of the decision space as a topographical map, and then picking the most favourable neighbour to move toward the goal.
B. Starts with an approximate solution and randomly swaps two elements, keeping the best selection as it moves toward the goal.
C. Divide and Conquer to split the decision space into smaller subproblems as it moves toward the goal
D. Escaping local optima by allowing some “bad” moves initially but gradually decreases the frequency of accepting “bad” moves as it works toward the goal.

Question 16

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 17

The hidden layers within an Artificial Neural Network:

A. Identify the errors in the predictions made by the Artificial Neural Network

B. Accept weighted inputs from previous layers and apply an activation function that is fed forward.

C. Apply random weightings to the input layer of information during training

D. Accept weighted information from the output layers.

Question 18

Given a rod of length N centimetres and a table of prices p[i] for i=1..N, determine the maximum revenue for cutting up the rod and selling the pieces.

- A rod of length i centimetres will be sold for price[i] dollars
- Cutting is free (simplifying assumption)

An example of a price table for different cutting lengths:

Length i	1	2	3	4	5	6	7	8	9	10
Price p[i]	1	5	8	9	10	17	17	20	24	30

A Dynamic programming solution has been partially defined in pseudocode for the rod-cutting problem.

```
Function CutRodDP(input: Array Pr, N)
// Pr is the price array, reference from 1
// N is the length of the uncut Rod
// Solution is the solution array, built from the bottom up, zero referenced
Solution[0..N]:=0  // initialise the solution array with zeros
For i=1 to N do
     For j=1 to i do
          If ((i-j) ≥ 0) then



          End if
      End do
End do
Return Solution[N]
End function
```

The missing lines of pseudocode that need to be inserted to complete the Dynamic programming solution are:

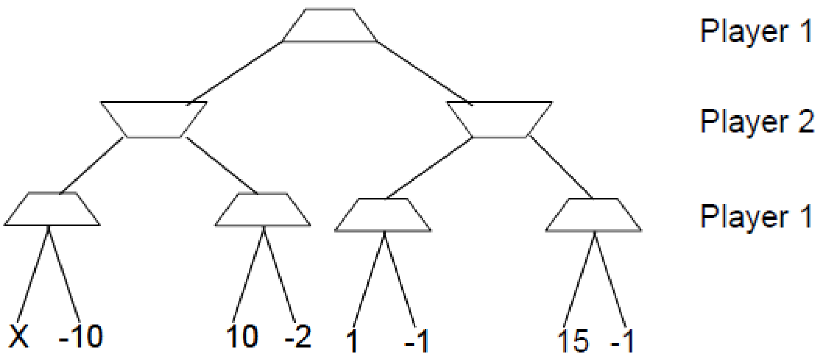
A.	`If (Pr[j] + Solution[i-j] < Solution[i]) then`
	`Solution[i] := Pr[j] + Solution[i-j]`
	`End if`

B.	`If (Pr[j] + Solution[i-j] > Solution[i]) then`
	`Solution[j] := Pr[i] + Solution[i-j]`
	`End if`

C.	`If (Pr[j] + Solution[i-j] > Solution[i]) then`
	`Solution[i] := Pr[j] + Solution[i+j]`
	`End if`

D.	`If (Pr[j] + Solution[i-j] > Solution[i]) then`
	`Solution[i] := Pr[j] + Solution[i-j]`
	`End if`

Question 19



The game tree above has an unknown payoff of *x*. Player 1 moves first and attempts to maximise the value of this play. If Player 2 is the minimising agent and is equally skilled (and Player 1 knows this). What possible values of *x* will be used if Player 1 ends up choosing the left action?

A. $1 < x < 10$

B. $x > -10$

C. $-10 < x < 10$

D. $x > 1$

Question 20

Which of the following problems is computable according to the Church Turing Thesis.

A. Halting Problem

B. Busy Beaver Problem

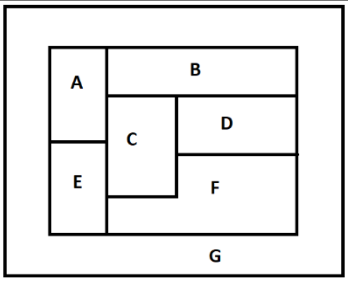
C. Hilbert's project to formalise mathematics.

D. Travelling Salesman Problem.

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

Question 1 (10 marks)

Consider the following map showing adjacent areas:



a. A cartographer wants to colour the map so that no two neighbouring countries have the same colour. Reduce the problem to a graph colouring problem by representing the map as a graph in the space above. (2 marks)

b. Add a method to the Abstract Data type signature for finding the neighbours of a node within a graph. (1 marks)

```
name graph;
import node, edge, list, int, boolean;
ops    newGraph   :  → graph;
       allNodes   : graph → list;
       allEdges   : graph → list;
       addNode    : graph × node → graph;
       deleteNode : graph × node → graph;
       NodeExists : graph × node → boolean;
       addEdge    : graph × edge → graph;
       deleteEdge : graph × edge → graph;
       EdgeExists : graph × edge → boolean;
```

c. Briefly explain the steps for the development of a greedy algorithm using the graph to try to solve this problem using the minimum number of colours. (2 marks)

d. What happens as the number of countries in the map becomes very large? Is a greedy algorithm strategy going to find the minimum colours required to colour? Explain. (2 marks)

Question 1 (Continued)

e. Describe a suitable approach for colouring a map of the 54 recognised countries of the continent of Africa. Justify and explain your approach and any limitations. (3 marks)

Question 2 (4 marks)

a. Write an efficient **iterative** algorithm in structured pseudocode to check whether two given words are anagrams of each other, that is whether one can be obtained from the other by permuting its letters. For example *gears* and *rages* are anagrams. (4 marks)

Question 3 (7 marks)

Consider the following pseudocode modules for the quicksort algorithm, that accepts as input an array A, and integer indexes, left and right into the array.
The **store** index is incremented by 1 with the action **store++** in the pseudocode below**.**

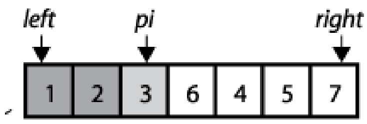
```
quickSort (A, left, right)
1.   if (left < right) then
2.      pi = partition (A, left, right)
3.      quickSort (A, left, pi – 1)
4.      quickSort (A, pi + 1, right)
end
```

```
partition (A, left, right)
1.   p = select pivot in A[left, right]
2.   swap A[p] and A[right]
3.   store = left
4.   for i = left to right – 1 do
5.      if (A[i] ≤ A[right]) then
6.         swap A[i] and A[store]
7.         store++
8.   swap A[store] and A[right]
9.   return store
end
```

a. What are the main design patterns used in the algorithm? Justify your response using the modules defined and identify the lines of pseudocode for each design pattern described. (2 marks)

b. Run the partition module shown above on the following data outlining the changes in the array when the pivot chosen is the value "3" in position 3 of the array A. (3 marks)

A							
	6	5	3	1	4	2	7

Identify the module and lines of code	Show the Input/Output array A contents, the left, pi, right and store index positions in the array A.
For example: Quicksort line 2 pi=partition(A,left,right)	

Question 3 (continued)

c. What is the best and worst case time complexity for the Quicksort algorithm shown? Justify your responses and support your conclusion by identifying the conditions or lines of code that lead to each of the cases. (2 marks)

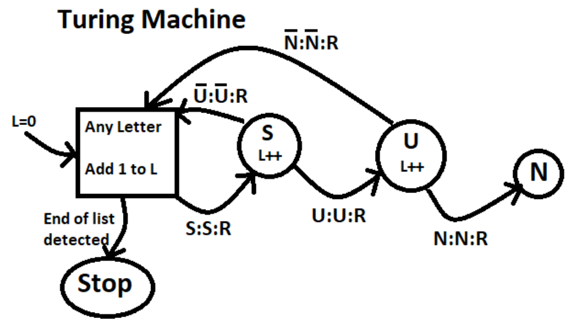
Question 4 (5 marks)

a. Explain how the Breadth-first search algorithm can be used to see if an undirected graph has any cycles. (2 marks)

b. Which of the two graph traversal algorithms Depth-first and Breadth-first will be able to find cycles faster in an undirected graph? (If there is no clear winner, give an example using a graph with four nodes A, B, C, D where one traversal algorithm is better, and another example where the other is better.) (3 marks)

Question 5 (13 marks)

Consider the following Turing Machine (TM) for detecting the sequence "SUN" in a list of sequential symbols.



Legend for the TM

L is a numeric variable

L++ increments the numeric variable by 1

Transitions on edges:

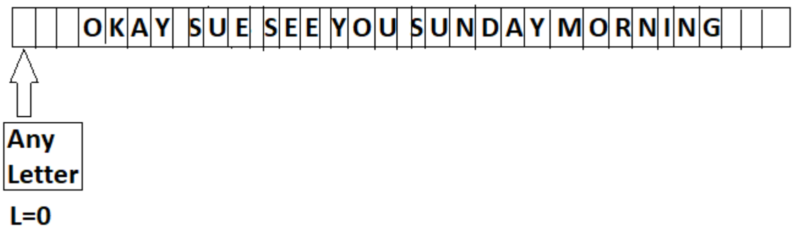
$InSymbol : OutSymbol : Movement$

Movement can be R (Right) or L (Left)

$S : S : R$, means if S input, then S output, then move 1 right

$\bar{S} : \bar{S} : R$, meant if not S input, then not S output, then move 1 right

a. What is the value in the variable L after executing the TM on the following tape of sequential symbols starting in the state shown and the arrow position? How many times is the TM in the state "S"? Justify your responses. (2 marks)



b. Translate this TM for finding "SUN" into structured pseudocode. (3 marks)

Question 5 (continued)

c. What is the best and worst case time complexity for this algorithm? Justify your responses. (2 marks)

d. What is computational complexity? Explain how computational complexity is related to the concept of a Turing Machine in general? (3 marks)

e. How is the concept of the Turing Machine related to Cobham's thesis and to the Church Turing thesis? What is the main difference in these theses? (3 marks)

Question 6 (12 marks)

When looking for the shortest path between any two nodes in a weighted connected graph G={V,E} we can use:

Dijkstra's using a minimum priority queue	**Bellman Ford**	**Floyd Warshall**

a. Complete the following table with the **main steps** used by each of these algorithms to find the shortest path from a source node.

Dijkstras (2 marks)	**Bellman Ford** (2 marks)	**Floyd Warshall** (2 marks)
	:	

b. What is the time complexity using each of these algorithms to find ***all*** the shortest pairs of node distances? Briefly justify your response with appropriate notation. (3 marks)

c. In which cases is it preferable to use each algorithm? Briefly justify your response. (3 marks)

Question 7 (9 marks)

a. What is PageRank? What is its purpose, and what information is used to determine it? (2 marks)

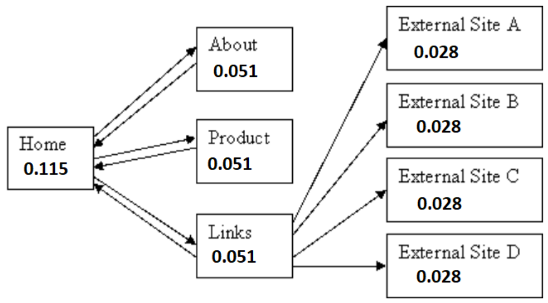
b. Complete the missing components of the following description of PageRank. (2 marks)

A page X in the web has other pages $\boldsymbol{T_1 \ldots T_N}$ *which point to it. The parameter d is a damping factor set to* ________, *which is the probability of following a* ________ *to another page.*
To initialise the algorithm Pr(X) is set to ________.
If L(X) is defined as the number or count of links going out of page X.
The PageRank of a page X is given as follows:

$$Pr(X) = \frac{(1-d)}{N} + d[Pr(\boldsymbol{T_1})/\underline{\qquad\qquad} + \ldots + Pr(\boldsymbol{T_N})/\underline{\qquad\qquad})$$

The sum of the PageRank of web pages in a system will be ________.

Lee calculated the PageRank for a system of 8 web pages and came up with the following results:



c. There is a problem with Lee's calculations. What is your diagnosis of the problem and how can it be fixed? (2 marks)

d. Show the recurrence formula(s) for finding the PageRank of the **Product** page in this system. (3 marks)

Question 8 (11 marks)

Consider a city centre where the streets are layed out on a grid parallel and perpendicular to each other. We are interested in walking from the upper left-hand corner of the grid to the lower right-hand corner. Unfortunately the city has bad neighbourhoods, whose intersections we do not want to walk through.

We are given an M × N matrix of BAD intersections where if BAD[i,j]=1 then this intersection is to be avoided and if BAD[i,j]=0, then this intersection is ok to walk through.

a. For a 2 × 3 street grid give an example of the contents of BAD such that there is no safe path from the upper left corner to the lower right corner. (1 mark)

b. Give an alternative model for this information and outline an algorithm that can be used to find a path across the grid that avoids bad neighbourhoods if one exists. Explain how the algorithm will work to find the path. (3 marks)

c. If the travelling one block in any direction is equal in cost, outline an algorithm that can find the shortest path across the grid that avoids bad neighbourhoods. (3 marks)

Question 8 (continued)

d. ~~Outline a process that could be used by the DNA Computation model to find the safest and shortest paths through the city.~~ ~~(2 marks)~~

e. How can you modify your solution for part c to restrict the traversal of each intersection once only? (2 marks)

Question 9 (9 marks)

Suppose you are choosing between the following two algorithms to solve a particular problem:

- ***Algorithm A*** *solves problems of size n by recursively solving two sub-problems of size (n-1) and then combining the solutions in constant time.*
- ***Algorithm B*** *solves problems of size n by dividing them into four sub-problems of size (n/2), recursively solving each sub-problem, and then combining the solutions in quadratic time.*

For algorithms A and B the base case when n=1 is solved in constant time.

a. What are the time recurrence relations for each of these algorithms expressed in T(n) notation? (2 marks)

b. What are the running times of each of these algorithms in big O notation, and which would you choose to solve the particular problem? (2 marks)

Another algorithm is presented that can solve this particular problem,
Algorithm C *solves the problem in* $O(1.001^n)$*)*.

c. Out of the algorithms A, B, and C, which one would you choose to solve the particular problem with considering different sizes of n? Justify and explain your selection. (3 marks)

d. Which algorithm will always terminate with the correct solution? Explain your selection. (2 marks)

END OF TRIAL EXAM 1