SOLUTIONS

ALGORITHMICS UNIT 3 & 4

CHES Trial Exam 1: 2023

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.

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

- the Master Theorem for solving recurrence relations of the form:

$$T(n) = \begin{cases} a \cdot T\left(\frac{n}{b}\right) + kn^c & \text{if } n > 1 \\ d & \text{if } n = 1 \end{cases}$$

where $a > 0, b > 1, c \geq 0, d \geq 0, k > 0$

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

The VCAA form of Master Theorem is equivalent to the form of Master Theorem taught in our class by consideration of log laws.

$$\log_b a = c \iff a = b^c \iff \frac{a}{b^c} = 1$$

$$\log_b a < c \iff a < b^c \iff \frac{a}{b^c} < 1$$

$$\log_b a > c \iff a > b^c \iff \frac{a}{b^c} > 1$$

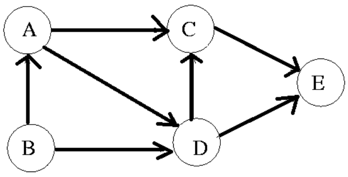
$$T(n) = aT\left(\frac{n}{b}\right) + f(n^k)$$

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

SECTION A – Multiple Choice – select one option only

Question 1

Consider the following Digraph:



A topological sort can be found because:

A. The directed graph is connected and only has directed acyclic components.

B. The directed graph is complete and has a cycle.

C. The directed graph is connected and has a cycle.

D. The directed graph is complete and has does not contain a cycle.

Question 2

The diameter or width of a graph G is:

A. the length of the shortest path between any two nodes

B. the largest distance between any pair of vertices. If G is disconnected, then its diameter is infinite.

C. the number of edges in a circuit, or cycle in the graph

D. a tour of a graph G which contains every edge of G.

Question 3

Which of the following is not a transparency issue in AI?

A. The ability to explain how an AI system makes decisions

B. The ability to access the data that an AI system is trained on

C. The ability to audit an AI system for bias

D. The ability to control the use of an AI system

Question 4

Q.enqueue(6)
Q.enqueue(3)
Q.dequeue( )
Q.enqueue(2)
Q.dequeue( )

Given an empty queue Q, what does Q look like after the operations above?

A. 6

B. 3

C. 2

D. It is empty

Question 5

What is the result of the following ABC recursive algorithm?

```
Function ABC(input number)
   If (number < 2) then
       Report number
   Else
       Report number + ABC(round(0.5*number))
   End if
End function
```

When called with ***ABC(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 6

Here are the steps involved in training a neural network using stochastic gradient descent:

a) Initialize the weights of the neural network to random values.
b) Randomly sample a batch of data from the training set.
c) Calculate the gradient of the loss function with respect to the weights of the neural network.
d) Update the weights of the neural network in the direction of the gradient.
e) Repeat steps 2-4 until the loss function converges to a minimum value.

What is the missing step 4?

A. Calibrate the weights of the neural network to previous values.

B. Randomly adjust the weights of the neural network to a cooling temperature schedule.

C. Backtrack to the global minimum of the loss function.

D. Update the weights of the neural network in the direction of the gradient.

Question 7

When considering different types of algorithms, which one of the following statements is **true**?

A. Divide and conquer algorithms are always faster than greedy algorithms for the same problem.

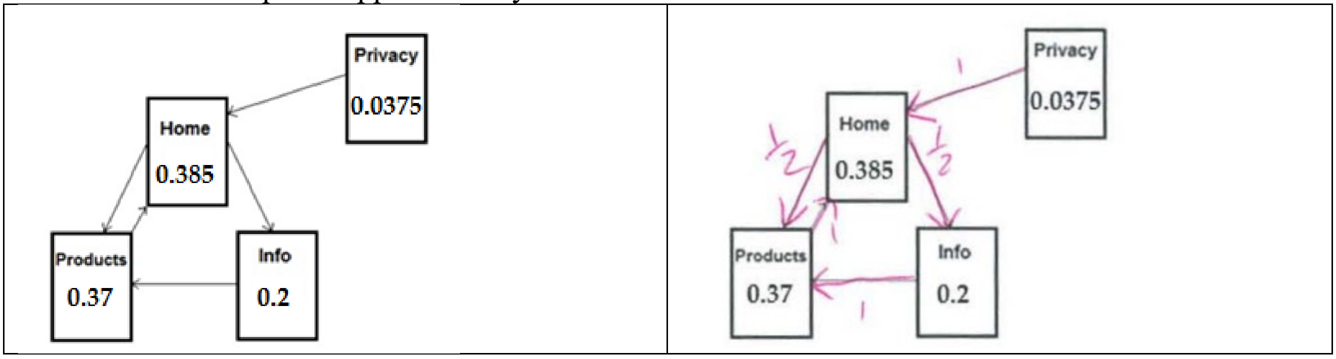
B. Greedy algorithms are always faster than divide and conquer algorithms for the same problem.

C. Greedy algorithms give good approximate answers to problems, but never the best possible answer.

D. Brute-force algorithms can never be faster than a well-designed greedy algorithm for the same problem.

Question 8

Consider the following web page system with an associated page rank value shown on each page. These values add up to 1 approximately.



The page rank for the **Products** page would be determined by which of the following recurrence formulas?

A. $\Pr(\text{Products}_{i+1}) = \frac{0.15}{4} + 0.85(\Pr(\text{Info}_i) + \Pr(\text{Home}_i)),\ \textit{where}\ \Pr(\text{Info}_0) = \Pr(\text{Home}_0) = \frac{1}{4}$

B. $\Pr(\text{Products}_{i+1}) = \frac{0.15}{4} + 0.85\left(\Pr(\text{Info}_i) + \frac{1}{2}\Pr(\text{Home}_i) + \frac{1}{4}\Pr(\text{Product}_i)\right),\ \textit{where}\ \Pr(\text{Info}_0) = \Pr(\text{Home}_0) = \Pr(\text{Product}_0) = \frac{1}{4}$

C. $\Pr(\text{Products}_{i+1}) = \frac{0.15}{4} + 0.85\left(\Pr(\text{Info}_i) + \frac{1}{3}\Pr(\text{Home}_i)\right),\ \textit{where}\ \Pr(\text{Info}_0) = \Pr(\text{Home}_0) = \frac{1}{3}$

D. $\Pr(\text{Products}_{i+1}) = \frac{0.15}{4} + 0.85\left(\Pr(\text{Info}_i) + \frac{1}{2}\Pr(\text{Home}_i)\right),\ \textit{where}\ \Pr(\text{Info}_0) = \Pr(\text{Home}_0) = \frac{1}{4}$

Question 9

Algorithm XYZ has the following worst time complexity $O(2^n)$. According to Cobham's thesis this algorithm is:

A. Tractable

B. Feasible

C. Non-computable

D. Intractable

Question 10

Uncomputable functions correspond to problems that are called:

A. Undecidable
B. Intractable
C. Polynomial time
D. none of these

Question 11

Dijkstra's single-source shortest path algorithm in an undirected graph reports distances from the source to each node. These distances ……

A. are the shortest possible distances to every destination node.
B. are never the shortest possible distances when negative edge weights are present.
C. may be the shortest possible distances when negative edge weights are present.
D. may not always be the shortest possible distances when all edge weights are positive.

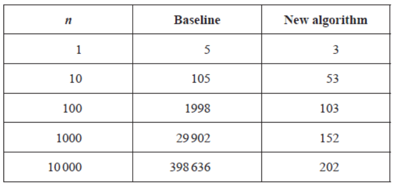
Question 12

What does it mean when we say that an algorithm X is asymptotically more efficient than Y?

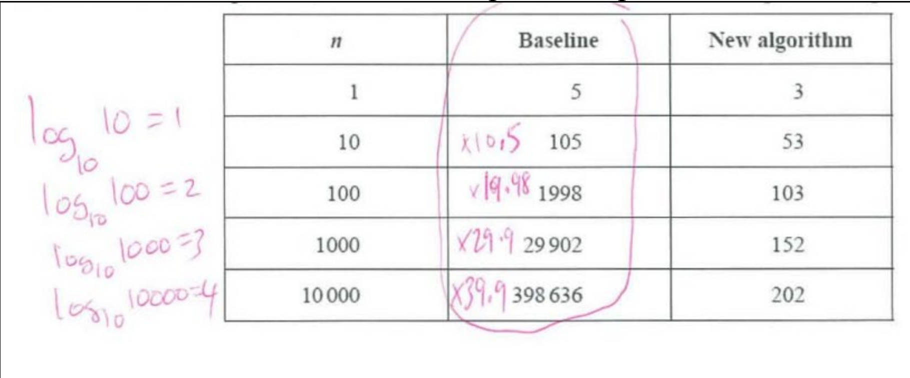
A. X will be a better choice for all inputs
B. X will be a better choice for all inputs except small inputs
C. X will be a better choice for all inputs except large inputs
D. Y will be a better choice for small inputs

Question 13

The following data shows running times (in microseconds) for a new algorithm compared with a standard baseline algorithm, where *n* is a measure of the size of the input to the problem.



n	Baseline	New algorithm
1	5	3
10	105	53
100	1998	103
1000	29 902	152
10 000	398 636	202



$\log_{10} 10 = 1$

$\log_{10} 100 = 2$

$\log_{10} 1000 = 3$

$\log_{10} 10000 = 4$

n	Baseline	New algorithm
1	5	3
10	x10.5 105	53
100	x19.98 1998	103
1000	x29.9 29 902	152
10 000	x39.9 398 636	202

Which Big-O expression most closely describes the running time of the baseline algorithm?

A. $O(n)$

B. $O(n^2)$

C. $O(\log n)$

D. $O(n \log n)$

Question 14

For Graph Colouring, we wish to assign the minimum number of colours, to the nodes, where neigbouring nodes cannot have the same colour, the following algorithm is using the Greedy strategy.

1. *Create a list of nodes in order of degree descending*
2. *Pick a new colour C*
3. *Go through the list, colouring each node not connected to coloured nodes with [………………..]*
4. *Remove the coloured nodes from the list and repeat from step 2 until all the list is empty*

<u>The missing action in the pseudocode is:</u>

A. A different colour

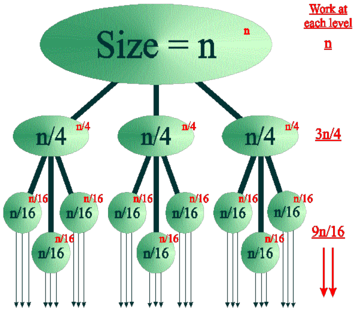
B. The same colour

C. A random colour

D. The colour C

Question 15

The call tree for a recursive divide & conquer algorithm is shown below:



The time complexity recurrence relation and the time complexity for this algorithm respectively are:

A. $T(n) = 4T(n/3) + f(n^1), O(n^{log(4)})$

B. $T(n) = 3T(n/4) + f(n^0), O(n^{log(3)})$

C. $T(n) = 3T(n/4) + f(n^1), O(n)$

D. $T(n) = 3T(n-1) + f(n^1), O(3^n)$

Question 16

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

A. Every recursive method must have a base case or a stopping condition.

B. Every recursive call reduces the original problem, bringing it increasingly closer to a base case until it becomes that case.

C. Infinite recursion can occur if recursion does not reduce the problem in a manner that allows it to eventually converge into the base case.

D. Every recursive method must have a return value.

Question 17

What is the missing pseudocode to complete the following function for computing factorials?

```
Function factorial(Input n)
   if (n is equal to 0) then
     return 1
   else
     return _____________
   end if
End function
```

A. n * (n - 1)

B. n

C. n * factorial(n - 1)

D. n + factorial(n - 1)

Question 18

Consider the following algorithms X and Y that have triple nested loops, what would the algorithm complexity be for each algorithm?

```
Algorithm X (input n)
Sum:=0
For i=1 to n do
   For j=1 to n do
      For k=1 to n do
          Sum:=Sum + 1
      End do
   End do
End do
End Algorithm
```

```
Algorithm Y (input n)
Sum:=0
For i=1 to n do
   For j=1 to n do
      For k=1 to n do
          Mergesort(an Array of size n)
      End do
   End do
End do
End Algorithm
```

A. Algorithm X has time complexity O(n^3), Algorithm Y has time complexity O(n^3)

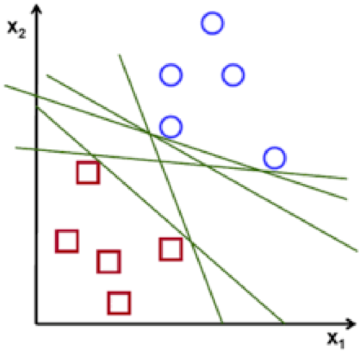
B. Algorithm X has time complexity O(n^3), Algorithm Y has time complexity O(n^4logn)

C. Algorithm X has time complexity O(n^3), Algorithm Y has time complexity O(n^4)

D. Algorithm X has time complexity O(n^3), Algorithm Y has time complexity O(n^3logn)

Question 19

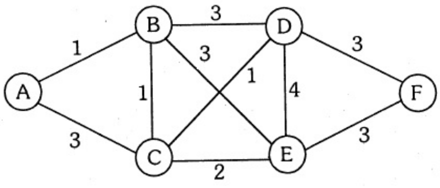
Where does a SVM draw its classification line using a linear kernel?



A. A best fit regression line through the data points
B. Equidistant between the support vectors identifying the cluster boundaries
C. Across the margin classifier of separation of the clusters
D. Through the maximum kernel function applied to the data points

Question 20

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

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

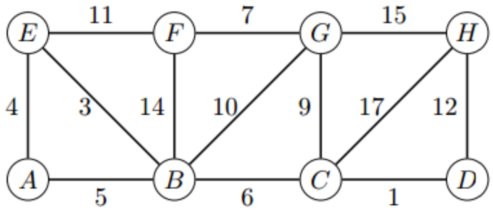
Question 1 (9 marks)

a. Explain the main difference between DFS, BFS and Best-First Search graph traversal algorithms, in particular discuss the difference in each algorithm structure and abstract data types used by each algorithm. (3 marks)

DFS - Depth uses a stack for processing nodes, explores deeply

BFS - Breadth uses a queue for processing nodes, fans out

Best first uses a priority queue where f(n) determines the ranking of nodes for processing.



b. Starting at node A, run the Depth First Search algorithm to node G on the graph shown, listing the order of nodes visited in alphabetic order where possible. (1 mark)______

A, B, C, D , H, G

c. Starting at node A, run the Breadth First Search algorithm to node G on the graph shown, listing the order of nodes visited in alphabetic order where possible. (1 mark)

A, B, E, C, F, G

d. Starting at node A, run the Best First Search algorithm using the edge weights as rankings to node G on the graph shown, listing the order of nodes visited in alphabetic order where possible. (1 mark)

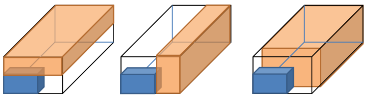
A, E, B, C, G

e. Describe how the A* Search algorithm would work to find a path from node A to node D using a heuristic cost stored on each node such as h(A), h(B), h(C)…etc, outlining any assumptions about the heuristic cost. (3 marks)

- A* search is a combination of Dijkstra's shortest path algorithm and Best First search.
- It sets the priority of nodes n to f(n) = g(n) + h(n) where g(n) is the shortest path so far from the source according to Dijkstra's and h(n) is the heuristic for setting the ranking of the node in the minimum priority queue for expansion
- A* will do better than Dijkstra's if the h(n) is less than or equal to the actual cost to the target node.

Question 2 (9 marks)

In the bin packing decision problem, objects of different volumes must be packed into a fixed number of bins each of given volume V1, V2, Vn.



a) In computational complexity theory this bin packing problem is classified as NP-Hard or NP-complete. Explain why this problem has two classifications and describe the circumstances for each of the classifications. ***(4 marks)***

- The optimal Bin packing problem is an optimisation problem that is NP-Hard, as it has factorial or exponential decisions for finding the optimum solution. It is similar in its structure and aims as the optimal Knapsack 01 problem, in that the bins have a capacity limit.
- For the optimal version of the Bin Packing even the solution cannot be check for correctness in polynomial time hence NP-Hard.
- For the decision version of the Bin packing problem the complexity is NP-Complete problem because the solution cannot be found in polynomial time. The decision version will determine some threshold has been reached that is satisfactory or not by linearly examining each item in the bin and keeping a tally to inform.
- So should a solution be found using a heuristic or other method it can be verified against the threshold in polynomial

b) Describe in structured pseudocode how a Simulated Annealing Heuristic algorithm could be set up to solve the Bin packing problem described above. (5 marks)

- Let X be a randomly selected subset of all the different object available for packing as the current subset.
- Create a helper function for returning a neighbour subset Y of subset X by Initialize Y to be a copy of the elements of X where a small amount of random objects are removed and/or added to make Y.
- While the temperature T is above threshold T_0 try to find the optimal $\sum$(elements in X) repeatedly do the following:
    - ○ Find a random neighbour Y of the current subset X If $\sum$(elements in Y) are better than $\sum$(elements in X), then make it the current subset X.
    - ○ Keep track of the best subset so far
    - ○ If $\sum$(elements in Y) is worse than $\sum$(elements in X), then if a random probability is less than the value of $e^{-\Delta/T}$, make Y the current subset, where Δ is the absolute value of the difference in sums between set X and Y.
    - ○ Reduce the temperature T

Question 3 (9 marks)

While clearly identifying the base case(s) and recurring case(s), define in structured pseudocode Decrease & Conquer recursive algorithms to calculate:

a) The sum of two integers m,n, here is the hybrid function defined in mathematical notation: (2 marks)

$$s(m,n) = \begin{cases} m & ,where\ n = 0 \\ 1 + s(m, n-1) & ,where\ n > 0 \end{cases}$$

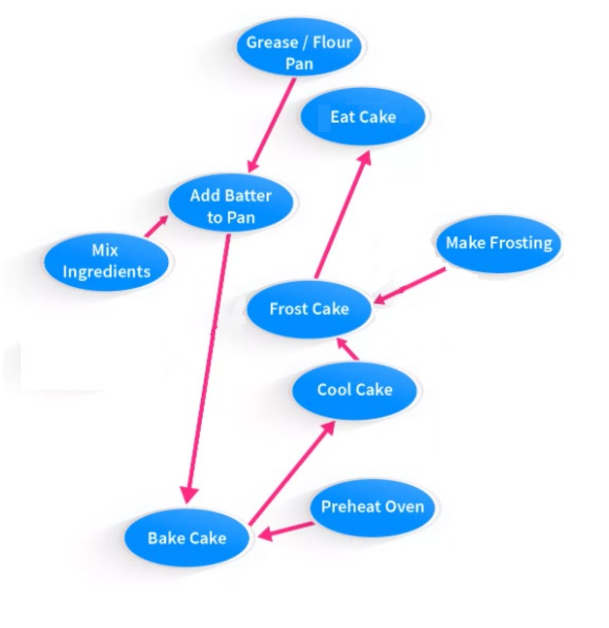
```
function Sum(m, n)
 if n = 0 then
    return m  // base case
  else
      return 1 + Sum(m, n-1)  // recursive case
  endif
end function
```

b) The power of m to n, where m is an integer m and n is a positive integer n. (3 marks)

$$power(m,n) = \begin{cases} 1 & ,where\ n = 0 \\ m & ,where\ n = 1 \\ m \times power(m, n-1), & where\ n > 0 \end{cases}$$

```
function Power(m, n)
  if (n = 0) then
    return 1 // base case
  else if n = 1 then
     return m  // base case
  else
     return m * Power(m, n-1)  // recursive case
   endif
end function
```

d) Create a clear proof by Induction showing the correctness of the recursive Topological Sort algorithm for input from directed graphs depicting dependant tasks. (4 marks)



```
Algorithm topologicalSort(G, sourceNodes)
// G input acyclic, directed graph
// sourceNodes a list that was initially empty
// Assumption: at least one source node in G
  if (|V|> 0) then
      foreach node n in G.V do
          if (in_degree(n) == 0) then
              print('Source Node ‘, n)
              append node n to sourceNodes
          end if
      end do
      Append SourceNodes to TopoList
      Remove nodes in sourceNodes from G // Layers
      topologicalSort(G, sourceNodes)    // recur on graph
  else
    print('Base Case Stop Recursion’)
  end if
End Algorithm
```

Step 1: Show the base case for P(1)

If P(1) = has to be a source

P(1-1) = 0 Base case

Step 2: Assume P(k) is correct for the algorithm

If P(k)

P(k-1) A new source node will be always be created when a node is removed

Continues recursion until...

P(1) = 0

Base case

Step 3: Show correctness for all cases P(k+1)

If P(k+1)

P(k)

P(k-1)...

We can assume we continue recursion due to our assumption...

P(1) = 0

Base case

New source node will be created after each recursion by taking the adjacent nodes to the source node. Once all nodes have been removed from the recursion, the algorithm will end. Proving the case for P(k+1) due to it fulfilling the base case and terminating.

Base Case (k = 1):

If there is only one node in the graph then its in-degree would be equal to 0. This means the node will be added to the sourceNodes list. This node will then be recorded and removed from the graph. The function would be called again with v = 0, hence terminating the recursion.

The output would be correct because the sorted graph will be identical to the original graph.

If there are isolated nodes in the graph G they will be discovered as their in-degree is equal to zero and added to sourceNodes and removed from G.

Assume kth recursion is correct and has removed all the source nodes at this level and added them to sourceNodes.

k + 1 recursion:

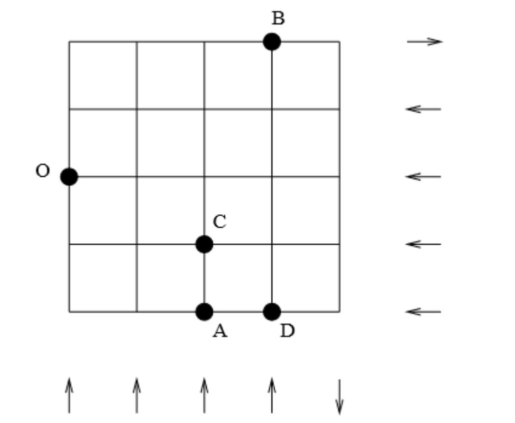
On the k+1 recursion all the child nodes that were adjacent to the source nodes of k th recursion now become the new source nodes and are removed along with their outgoing edges.

Termination:

As all isolated and connected nodes will in turn be removed from the graph the algorithm will terminate as the main condition |V| > 0 will become false.

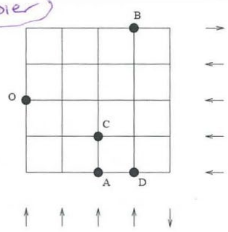
Question 4 (7 marks)

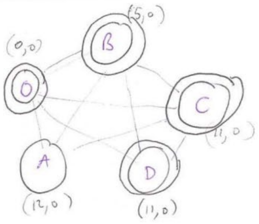
a) Below is a map of Oneway City, a poorly planned town with one-way streets as indicated. Represent the information for Oneway City using fit for purpose abstract data type. (2 marks)



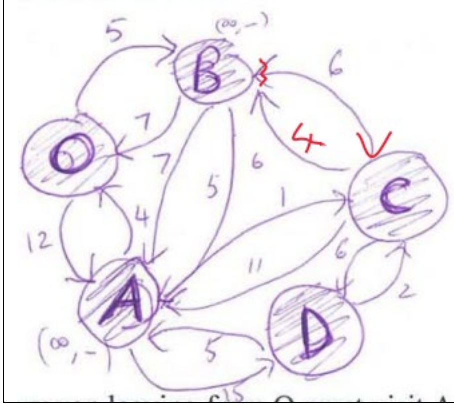
An adjacency matrix is easier

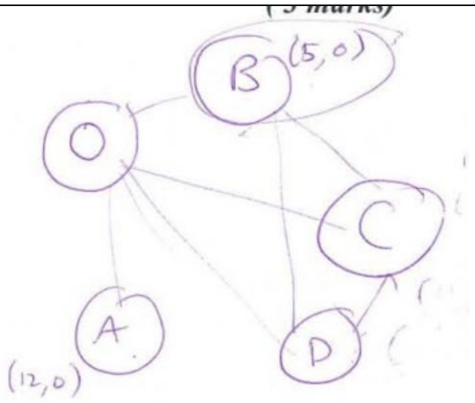
$$\text{from}\quad \begin{array}{c|ccccc} & O & A & B & C & D \\ \hline O & 0 & 12 & 5 & 11 & 11 \\ A & 4 & 0 & 5 & 1 & 15 \\ B & 7 & 7 & 0 & 6 & 6 \\ C & 3 & 11 & 4 & 0 & 14 \\ D & 5 & 1 & 4 & 2 & 0 \end{array}$$





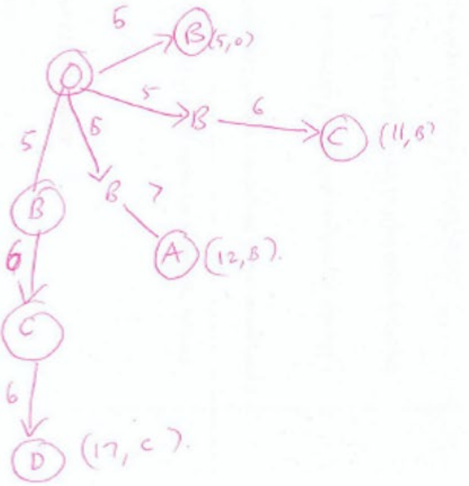
b) Draw the complete directed graph with vertices corresponding to O, A, B, and C (ignore D for now) and two edges between each pair of vertices, one in each direction. Label each with the distance (number of blocks) one must travel to get from one location to the other. For example, the distance from A to C is 1 block and from C to A is 11 blocks. ***(2 marks)***





c) Suppose a delivery van leaving from O visits locations A, B, C, and D. Use Dijkstra's algorithm to find the shortest route from O to any of A, B, C and D. Show the workings that you calculated together with predecessor nodes in finding this route. ***(3 marks)***

		A	~~B~~	C	D
	O	12_O	$\cancel{5}_O$	∞_O	∞_O
5	↓				
	B	12_B		$\cancel{11}_B$	∞_O
6	↓				
	C				17_C



Question 5 (12 marks)

a) What are the main differences between solving a problem using a traditional algorithm and using a neural network? (2 marks)

- Traditional algorithms need to be designed using sequence, iteration and conditional actions with deterministic methods to solve the problem
- A neural network learns to make predictions on the output based on learning from training data and feedback on success in training to form a mathematical relationship between inputs and outputs.

b) What are the main components of designing neural networks? (3 marks)

- Input layers have the input feature metrics and weights to pass to the hidden layer.
- Hidden layers apply activation functions to inputs from previous layers and pass onto possibly other hidden layers or output layers
- Output layers transmit the solution and predictions of output based on the inputs

c) What type of Machine Learning is a Neural Network? Justify your response. (2 marks)

- Neural networks are a supervised Machine Learning algorithm
- Learns by many examples of labelled data during training to adjust the weightings on the inputs to match the label.

d) Describe how a neural network is trained and what methods are used to minimise errors and then how are they used to solve problems (2 marks)

- Using the labelled data during training the Neural network "learns" the weightings by minimising the loss or error function in prediction vs actual using mathematical methods such as gradient descent or stochastic gradient descent.
- Once trained a Neural Network is limited to a particular kind of problem and can give output for new input feature data for that problem.

e) A neural network has been trained to understand written Chinese and respond to questions using written Chinese? Discuss if there would be any implications for Searle's views on Artificial intelligence by referencing the neural network model. (3 marks)

- Searle would argue that it is no different to the Chinese Room Thought Experiment where the room has been replaced by the neural network.
- While the neural network has been trained to read and interpret chinese characters and output with correct responses, it doesn't actually understand any Chinese and is simply trained to looking up rules in a similar way to the person in the Chinese Room would who doesn't understand Chinese.
- This will not change any of Searle's views on the impossibility of strong AI.

Question 6 (8 marks)

a. The following Quicksort algorithm calls the partition module to sort data. Complete the missing lines of the algorithms. (2 marks)

```
Algorithm Quicksort(A, low, high)
// Input A, array of numbers
// Input low, an index into the array A
// Input high, an index into the array A
    if (low < high) then
        p := Partition(A, low, high)
        Quicksort(A, low, p – 1)
        Quicksort(A,p+1, high)
     end if
End Algorithm
```

```
Algorithm Partition(A, low, high)
// Input A, array of numbers
// Input low, an index into the array A
// Input high, an index into the array A
    pivot := A[high]
    i := lo - 1
    for j := low to high - 1 do
        if A[j] < pivot then
            i := i + 1
            swap A[i] with A[j]
        end if
    end do
// move the pivot to the correct position
    i := i + 1
   swap A[high] with A[i]
   return i  // pivot index
End Algorithm
```

b. What design pattern is used by the Quicksort algorithm? Justify and describe which parts of the algorithm inform about the design pattern used. (2 marks)

- Divide and Conquer is used with two recursive calls after the partition step that puts the pivot p in the correct position and elements less that p before p, and greater than p after p in the array.
- The first recursive call is in values less than the pivot and the second recursive call is on values greater than the pivot.

c. What is the time complexity of the helper Partition function? Justify your response. (2 marks)

- The Partition helper function has linear time complexity O(n)
- since it has one for loop that goes through each element of the input array A.

d. What is the best case and worst case time and space complexity for the algorithm shown? Justify your responses using appropriate time complexity recurrence relations. (3 marks)

- In the best case the pivot will split the original unsorted data into equal halves for passing to the first recursive call to sort the lower half and the second recursive call to sort the upper half, which will result in $T(n)=T(n/2)+T(n/2)+n$, $T(1)=1$ recurrence relation and O(nlogn) time complexity.
- In the worst case the pivot p splits the data 1, (n-1) or (n-1),1 whith effectively becomes decrease and conquer recursion where the work outside the recursion is linear as done by the partition helper function, hence $T(n)=T(n-1)+n$, $T(1)=1$ time complexity recurrence relation and O(n^2) overall time complexity.

Question 7 (10 marks)

A Latin square is an array of k elements by k elements with some values already set.

For some Latin squares the goal is to have the integers 1 to k appear once in each row and once in each column.

Two examples of a completed k=5 numeric Latin Square is shown at right.

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 5 & 1 & 4 \\ 3 & 5 & 4 & 2 & 1 \\ 4 & 1 & 2 & 5 & 3 \\ 5 & 4 & 1 & 3 & 2 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 4 & 1 & 5 & 3 \\ 3 & 5 & 4 & 2 & 1 \\ 4 & 1 & 5 & 3 & 2 \\ 5 & 3 & 2 & 1 & 4 \end{bmatrix}$$

a) Write a clearly defined modular algorithm using **structured pseudocode with one or more helper function(s)** to validate and check that a completed Latin square provided as input of dimension k by k has been completed correctly and has the digits 1..k in each row and also in each column once only. (4 marks)

```
Function checkRow(Lsquare,row,k)
...create list check={1,2,….k}
  error = False
  for j=1 to k do
    if Lsquare[row,j] in check then
       remove j from check
    else
       error = True
  If check is not empty then
       error = True
  return error
end function

Function checkColumn(Lsquare,col,k)
...create list check={1,2,….k}
  error = False
  for j=1 to k do
    if Lsquare[j,col] in check then
       remove j from check
    else
       error = True
  If check is not empty then
       error = True
  return error
end function }
```

```
Function checkLatinSquare(Lsquare,k)
  Set e = 0
  For j = 1 to k do
    If (checkrow(Lsquare,j,k) == False) then
        e = e + 1
    If (checkcol(Lsquare,j,k) == False) then
        e = e + 1
  if (e is zero) then
      report correct Latin Square
  else
      report the number of errors is: e
end function.
```

NOTE: Answers will vary

b) What is the time complexity of the algorithm that you have defined in part a)? Justify your answer with reference to part a). (2 marks)

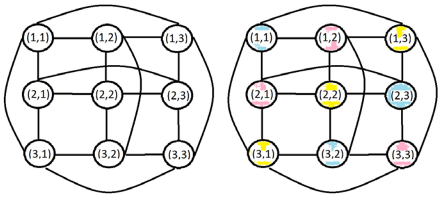
- $O(k^2)$ where k is the dimensions $k \times k$ of the Latin Square provided as the input.
- Justification of $O(k^2)$: as there will be $k \times (k + k) + c = 2k^2 + c$ basic actions, where c is constant as the main function checkLatinSquare has a for loop 1..k that calls checkRow which is $O(k)$ and also calls checkColumn which is $O(k)$, hence $O(k^2)$ for the overall algorithm.

Question 7 (continued)

c) Describe and demonstrate how solving a 3 x 3 Latin square problem can be set up and completed by filling in the digits 1,2,3 as a minimum colour graph colouring problem. (2 marks)

Let each cell of the Latin square be a node, then connect the nodes that are not allowed to have the same number within them by an edge according to the Latin Square rule.
Using the minimum colours – then colour the nodes such that no two adjacent nodes have the same colour. The solution to this problem represents the solution to the Latin Square. This should be colourable by 3 colours representing the digits 1,2,3.

(1,1)	(1,2)	(1,3)
(2,1)	(2,2)	(2,3)
(3,1)	(3,2)	(3,3)

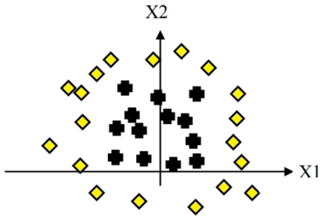


d) What is the complexity class of completing a Latin Square puzzle problem with digits 1..k? Justify your response. (2 marks)

The complexity class is NP-Complete as the puzzle cannot be solved in polynomial time, whereas a solution provided can be verified for correctness in polynomial time.
The reason it cannot be solved in polynomial time is the combinations for placing numbers in progress to a complete solution is factorial.

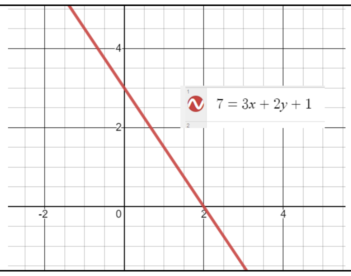
Question 8 (8 marks)

Suppose that we want to classify two dimensional data (i.e., X = [x1, x2]) into two classes: diamonds and crosses. We have a set of training data that is plotted as follows:



a) Can a single perceptron separate the data for the diamonds and the crosses? Explain why or why not and justify your response. (2 marks)

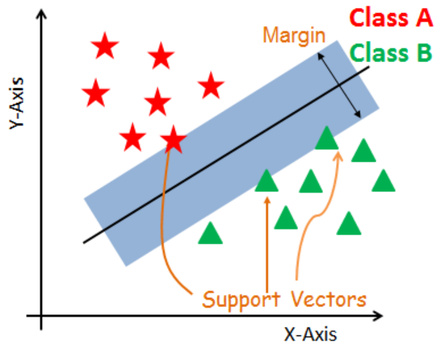
A single perceptron cannot linearly separate diamonds and crosses for the inputs X1 and X2 as the output equation for the perceptron O=aX1+bX2+X0 can only be a straight line.
For example as shown in the online graphing calculator Desmos:
https://www.desmos.com/calculator



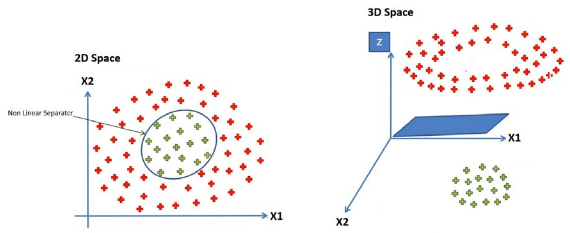
b) Describe the Support Vector Machine algorithm used in Artificial Intelligence. (2 marks)

Support Vector Machines is considered to be a classification approach, it but can be employed in both types of classification and regression problems. It can easily handle multiple continuous and categorical variables. SVM constructs a hyperplane in multidimensional space to separate different classes. SVM generates optimal hyperplane in an iterative manner, which is used to minimize an error. The core idea of SVM is to find a maximum marginal hyperplane(MMH) that best divides the dataset into classes.

c) Draw a labelled diagram representing the components of the Support Vector Machine. (2 marks)



d) Can a Support Vector Machine (SVM) machine learning algorithm classify this data from part a)? Explain why or why not and justify your response (diagrams accepted). (2 marks)

An SVM can apply a transformation known as a kernel trick to add another dimension to the data to allow it to be separated by a hyperplane in 3D. By applying the transformation $z=(X1-a)^2+(X2-b)^2$ and then using the 3rd dimension value $z <$ constant (radius) the crosses can be separated from the diamonds.	 2D Space X2 Non Linear Separator X1 3D Space Z X1 X2

Question 9 (8 marks)

Scenario 1: A company is behind schedule and over budget in building an AI application. The AI uses facial recognition to unlock a smartphone. Through testing the company found that the AI worked on **most** people's faces.

a) What ethical issues arise if the company goes ahead and decides to sell the phone and fix the AI problems in the next version of the phone? (2 marks)

Facial recognition systems are becoming more common in applications and can impact people. These systems are vulnerable to errors introduced during training by its human creators. This is referred to as bias. In this example, the data is biased towards certain people while excluding others. This scenario raises the issues of fairness and human-centred values such as diversity.

Scenario 2: An employee working for a company on an AI project finds out that the AI application could be hacked and used for criminal purposes.
The manager of the project instructs the employee to ignore it, saying 'Don't worry, that won't happen!'

b) What ethical issues arise in Scenario 2? What are the potential impacts that can arise? (2 marks)

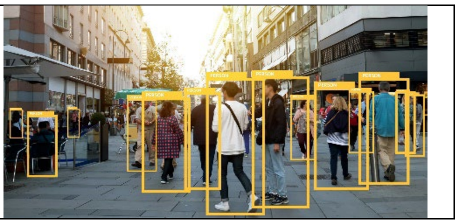
Accountability and responsibility of the creators of the AI is called into question in this scenario, the AI company is not behaving ethically if their system is knowingly made available as the customer privacy will be compromised if their information is hacked and used. Customers can rightly hold the creators liable for breach of privacy for use of this AI.

Scenario 3: The artwork has been generated using AI and is called "Melbourne skyline in the style of Picasso and Van Gogh" it is being printed on tourist merchandise and sold for commercial gain. Van Gogh work is in the public domain. To reproduce and/or depict works by Picasso, you will need the prior written consent of the Picasso Administration, the company representing the artist's rights holders.

c) What are the ethical issues that arise in Scenario 3? (2 marks)

Generative AI can plagiarise the intellectual work of artists without attribution for commercial gain, demonstrating a potential lack of responsibility for usage rights and fair compensation. The accountability of Generative AI is an area of ethical consideration as it mimics work in the style of another artist and infringes their intellectual property rights with just a few prompt words.

Scenario 4: A university campus has installed a AI surveillance system that monitors activities in various public areas of the campus to detect intruders, unauthorised visitors and combat petty crime and graffiti.



d) What are the benefits and drawbacks arising from the use of AI in Scenario 4? (2 marks)

AI surveillance systems have the ability to process vast amounts of data, enabling them to detect anomalies quickly. This can lead to more effective security measures, as AI systems can identify potential threats and alert authorities more quickly than traditional methods. On the negative sides the biases in facial recognition technology have already led to injustices using AI systems and the potential for breaches of privacy need to be reduced.

END OF TRIAL EXAM 1