Name:

ALGORITHMICS UNIT 3 & 4

Trial Exam 2: 2021

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		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 25 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.

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

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

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

Scratch	Python
set shopping to list add milk to shopping add bread to shopping add butter to shopping add eggs to shopping set shopping to quicksort shopping say shopping contains quinoa for 2 secs delete 3 of shopping replace item 2 of shopping with low fat spread say length of shopping for 2 secs	shopping=[] shopping.append('milk') shopping.append('bread') shopping.append('butter') shopping.append('eggs') shopping.sort() print('quinoa' in shopping) shopping.remove('eggs') shopping[shopping.index('butter')]='low fat spread' print (len(shopping))

What is the printed output from the ADT operations shown above in Scratch and in Python?

A. false, 3

B. quinoa, 3

C. false, 4

D. true, 3

Question 2

name Dictionary;
import key, *value*;
ops *newDictionary* : $\rightarrow$ [______];
insertDictionary : *key* $\times$ *value* $\times$ *dictionary* $\rightarrow$ *dictionary*;
removeDictionary : [______] $\times$ *dictionary* $\rightarrow$ *dictionary*;
lookupDictionary : *key* $\times$ *dictionary* $\rightarrow$ [______];

The missing terms for defining the Dictionary ADT in order are:

A. Dictionary, value, value

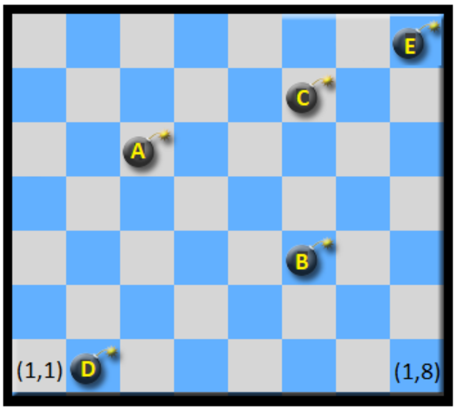
B. Dictionary, key, key

C. Dictionary, key, dictionary

D. Dictionary, key, value

Question 3

The following algorithm `Countit` counts how many bombs on a game board of n by m cells, as shown in the diagram, where cell (1,1) and cell (1,8) are shown on the **same row**.




```
Algorithm Countit(n,m)
Set numBombs to 0
For row = 1 to n
     For column = 1 to m
         If (gameBoard(row, column) is a bomb) then
               Add 1 to numBombs
         End if
     End for
End for
Print numBombs
End Algorithm
```

In what order will the bombs be discovered by **`Countit(7,8)`**?

A. E, C, A, B, D

B. D, B, A, C, E

C. D, A, B, C, E

D. A, B, C, D, E

Question 4

The time complexity of the Countit above in Question 3 is best described by:

A. O(n)

B. O(n+m)

C. O(nm)

D. O(n^2)

Question 5

The Halting problem is an example of:

A. A decidable problem

B. An undecidable problem

C. A complete problem

D. A trackable problem

Question 6

Dijkstra's algorithm is based on which design pattern?

A. Greedy design pattern

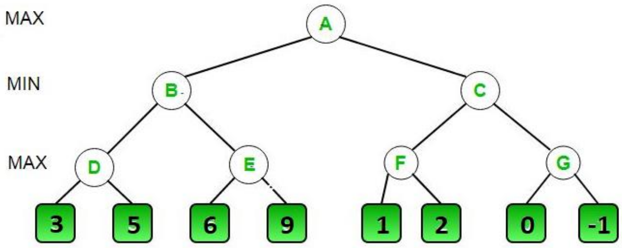
B. Backtracking design pattern

C. Dynamic Programming design pattern

D. Divide and Conquer design pattern

Question 7

~~Running the minimax algorithm on the game tree shown at the right will result in a score for the maximising player at node A of:~~



~~**A.** 6~~

~~**B.** 9~~

~~**C.** 8~~

~~**D.** 5~~

Question 8

Consider the following statements:

<u>Statement 1:</u> The basic idea of dynamic programming considers all possible cases and is essentially the opposite of a greedy strategy.
<u>Statement 2:</u> When dynamic programming is applied to a problem, it takes far less time as compared to other methods that don't take advantage of overlapping sub-problems.

Then

A. Only Statement 1 is true

B. Only Statement 2 is true

C. Both Statements are true

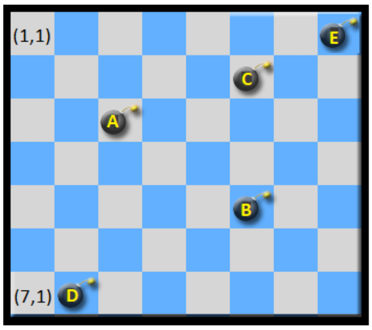
D. Both Statements are false

The following algorithm `CountBombs` is to be used to answer Question 9 and Question 10

```
Algorithm CountBombs(Board,n,m)
// Input Board grid of n rows and m columns
New Array cell:=[1,1] //[row,col]
NewQueue Q
Enqueue cell to Q
While Q is not empty do
    Set cell to Dequeue Q
    row := cell[1]
    col := cell[2]
    Set Board[row,col] as visited
    If ((row+1)≤n) AND NOT (Board[row+1,col] visited) then
       cell := [row+1,col]
       Enqueue cell to Q
    End if
    If (((row+1)≤n) AND ((col+1)≤m) AND NOT (Board[row+1,col+1] visited) then
       cell := [row+1,col+1]
       Enqueue cell to Q
    End if
    If ((col+1)≤m) AND NOT (Board[row,col+1] visited) then
       cell := [row,col+1]
       Enqueue cell to Q
    End if
End do
End Algorithm
```

Question 9

Consider the game board shown of rows by columns, cell (1,1) and cell (7,1) are shown in the same column.
Several bombs labelled A, B, C, D, E are located on the game board in distinct cells of the board.



In what order will the bombs be discovered by the algorithm defined above `Algorithm CountBombs(Board,7,8)`?

A. A, B, C, D, E

B. E, C, A, B, D

C. D, A, B, C, E

D. E, C, B, A, D

Question 10

Which graph traversal algorithm does CountBombs most closely follow?

A. Depth First Search

B. Best First Search

C. Random Search

D. Breadth First Search

Question 11

Consider the following recursive algorithm for Depth First Search defined in pseudocode as Function DFS below.
First of all, the visited array is initialised with false values. The use of the visited array determines which nodes have been visited to prevent the algorithm from visiting the same node more than once.

Which are the missing parts of the Function DFS?

```
Algorithm 1: Recursive DFS
  Data: G: The graph
        root: The starting node
  Result: Prints all nodes inside the graph in the DFS order
  visited ← {false};
  DFS(root);

  Function DFS(u):
    if visited[u] = ____ then
      return;
    end
    print(u);
    visited[u] ← true;
    for v ∈ G[u].neighbors() do
      ____________
    end
  end
```

A. *true, DFS(v)*

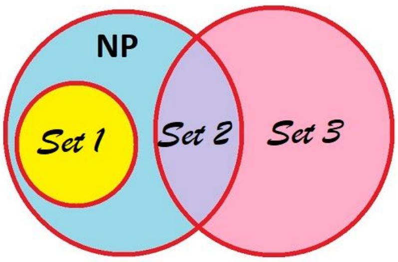
B. *false, DFS(u)*

C. *true, DFS(u)*

D. *false, DFS(v)*

Question 12

A partial diagram of the time complexity classes relationship is shown.



Set 1, Set 2 and Set 3 in consecutive order are:

A. NP-Complete, NP, NP-Hard

B. NP-Complete, NP-Hard, P

C. P, NP-Hard, NP-Complete

D. P, NP-Complete, NP-Hard

Question 13

The problem of opening the lock shown at the right in the diagram belongs to the complexity class:

A. P

B. NP

C. NP-Hard

D. Undecidable

Question 14

Which of the following best describes the position of Strong AI?

A. The principal value of computers is that they are powerful tools for studying the mind

B. Having a mind is a matter of having the right outputs

C. Computers cannot be minds

D. An appropriately programmed computer is a mind, in the sense that it can understand

Question 15

Which of the following best describes the “Systems reply” to Searle's thought experiment?

A. We only attribute understanding to people because of their behaviour, so we should for machines too

B. While the individual in the room doesn't understand the story, the system she's a part of does

C. There would be understanding if we put the system into a mechanism that walked around, perceiving

D. None of the above

Question 16

Consider the following algorithm X, what is the time complexity?

```
Algorithm X(n)
// input n an integer > 0
i:=n
sum:=0
while (i > 0) do
    j:=1
    while (j < n) do
        k:=0
        while (k < n) do
            sum:=sum + (i+j*k)
            k:=k+2
        end do
        j:=j*2
    end do
    i:=floor(i/2) // the floor function always rounds down
end do
end algorithm
```

A. $O(n^3)$

B. O(nlogn)

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

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

Question 17

The time complexity of a recursive algorithm with the number of actions based on the input size of n is given by $T(n) = 3T\left(\frac{n}{3}\right) + \frac{n}{2},\ T(1) = 1.$ What is the time complexity represented as a function of n?

A. O(n)

B. $O(n^2)$

C. O(nlogn)

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

Question 18

Let X be a problem that belongs to the class NP. Then which one of the following is TRUE?

A. There is no polynomial time algorithm for X.

B. If X can be solved deterministically in polynomial time, then P=NP.

C. If X is NP-Hard, then it is NP-Complete

D. X may be undecidable

Question 19

Let G be a complete undirected graph on 6 nodes. If the nodes of G are labelled, then the number of distinct cycles of length 4 in G is equal to:

A. 15

B. 30

C. 45

D. 360

Question 20

What does the following **mystery** algorithm defined in pseudocode below print when called as **mystery(3,4,5)**?

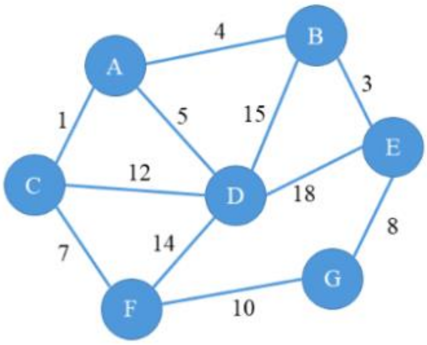
```
Algorithm mystery(n,a,b)
// Input n an integer
// input a an integer
// input b an integer

    if (n > 0) then
        mystery(n-1, a, b+n)  // 1st recursion
        print n,a,b //prints out variables
        mystery(n-1, b, a+n)  // 2nd recursion
    end if
end algorithm
```

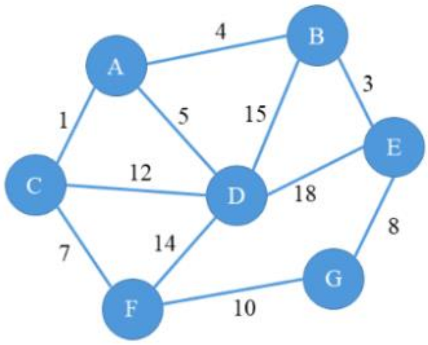
A		C	
	1 4 10 2 4 8 1 8 6 3 4 5 1 5 9 2 5 7 1 7 7		3 4 5 1 4 10 2 4 8 1 8 6 1 5 9 2 5 7 1 7 7
B	1 4 10 2 4 8 1 8 6 3 4 5	**D**	3 4 5 1 5 9 2 5 7 1 7 7

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

Question 1 (10 marks)

Consider the graph below shown in duplicate to answer parts a) and b)





a) List the order of edges that are added to the minimum spanning tree that Prim's algorithm would find if it used vertex A as the starting node. (2 marks)

b) List the order of edges that are added to the minimum spanning tree that Prim's algorithm would find it it used vertex G as the starting node. (2 marks)

c) In what circumstances is the minimum spanning tree of any undirected weighted graph unique? Justify your claims using a logical contradiction argument. (3 marks)

Question 1 (Continued)

d) In what circumstances are there multiple minimum spanning trees in an undirected weighted graph? Justify your claims. (2 marks)

e) What changes can be made to the graph above to make it have a non-unique minimum spanning tree? (1 mark)

Question 2 (6 marks)

A directed graph of 4 nodes and 6 directed edges is shown in Figure 1.

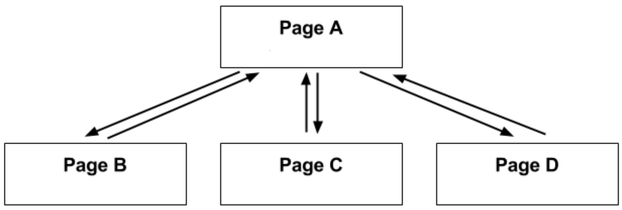


Figure 1

a) Calculate the Page ranks after initialisation for the directed graph in Figure 1 . (1 mark)

b) Calculate the Page ranks for Page A and Page D after iteration 1 for Figure 1. (2 marks)

Question 2 (Continued)

The navigation ***is changed*** for Page D in the directed graph as shown in Figure 2.

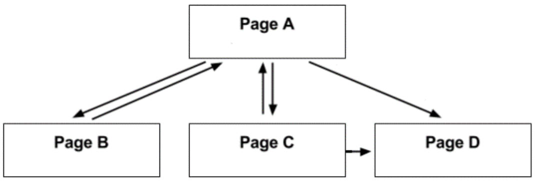


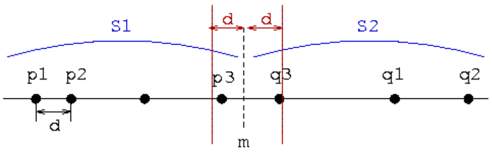
Figure 2

c) Explain how this change affects the Page rank calculations for Figure 2. (1 mark)

d) Show the Page Rank recurrence relation for A and B for the new graph shown in Figure 2 (2 marks)

Question 3 (12 marks)

Consider a set of points *S* on a number line, our goal is to determine which two of these points are minimally distant from each other.

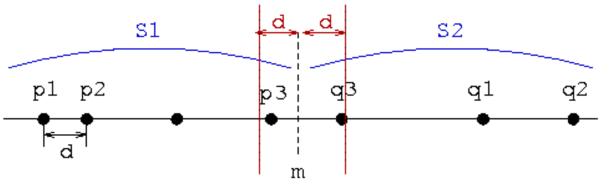


a) Outline a Naïve Brute Force approach for solving this problem. What would be the time complexity of the Naïve Brute Force approach? (2 marks)

b) What design pattern could be used to improve the time complexity of a Brute Force approach for this problem? (HINT: in the diagram above) Justify and state the improved time complexity. (3 marks)

Question 3 (continued)

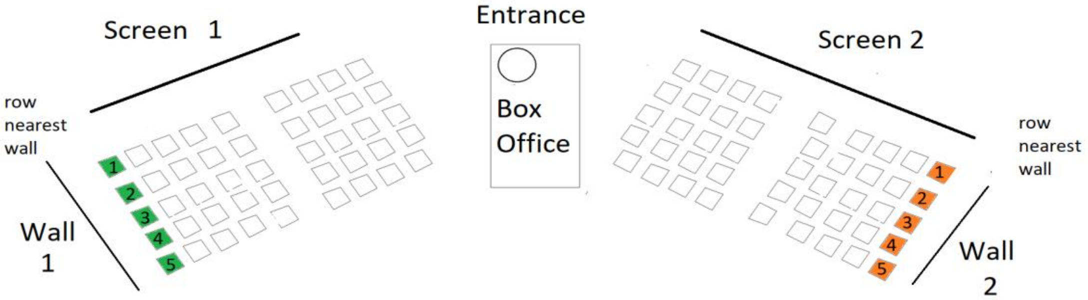
c) Outline an algorithm in plain English for solving this problem using the design pattern you have selected in part b) (3 marks)



d) Outline an algorithm in structured pseudocode for solving the closest pair of points 1D problem using the design pattern you have selected in part b) (4 marks)

Question 4 (14 marks)

Funkytown has a 2 Screen Drive-In where people watch movies on huge screens from their cars.



Car Entry Rules of Funkytown Drive-In (2 Screens, 2 Walls, 2 Parking Bays (each of 5×9 car spots))

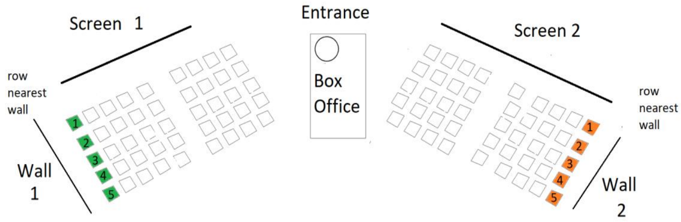
- Cars come into the entrance and are served at the box office in a first come first served order for a double session.
- Depending on their movie selection if there are still parking spots available then, each car is directed to the single lane entry for the parking spaces of either Screen 1 or Screen 2 (see diagram above).
- Cars are offered the other movie subject to availability if their first choice is not available, otherwise they are turned away.
- As cars make their way single file into the screen parking area, they must go to the empty rows nearest the wall filling the 5 spots in order from nearest the screen to the back before starting a new row, as shown in the diagram above.

a) Explain and justify what specific Abstract Data Types ADT(s) can be used to model the parking of vehicles for a double screening session according to the Funkytown Drive-In rules above. (4 marks)

b) Show the ADT operations to create, add/remove elements for the ADT(s) you have specified in part a) according to ADT signature definitions. (4 marks)

Question 4 (continued)

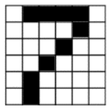
Refer to the Entry Rules of Funkytown Drive-In to answer this question.



c) Implement the car parking allocation **from the cars already queued at the Entrance gate** following the given rules using a **modular** and well structured algorithm written in pseudocode, using your ADT(s) from parts a) and assuming the ADT(s) from part b) operations have already been created. (6 marks)

Question 5 (10 marks)

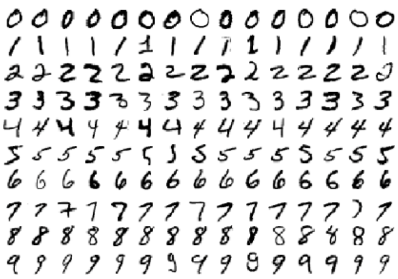
Consider the handwritten number “7” on a 6 by 6 pixel matrix image as shown.

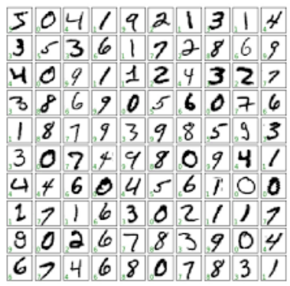


a) Describe in general terms how a Neural Network could have its inputs set up to recognise a handwritten “7” from the digital images of 6 by 6 pixel matrix images as shown. Show a diagrams to aid the description. (2 marks)

b) Describe in general terms the structure of a simple forward propagating Neural Network and how the Neural Network proceeds after it accepts the inputs and the actions that it does. (2 marks)

The handwriting Neural Network is provided with training data for many handwritten “7” s as well as other handwritten digits as 6 by 6 pixel images and uses back-propagation for training.





c) In general terms describe what is meant by training the Neural Network and using back-propagation for training? (2 marks)

Question 5 (continued)

Neural Networks computers/algorithms can learn how to read handwritten texts through training.

d) Describe in your own words how a Neural Network could display "strong AI" and "weak AI" through learning how to read handwritten texts. (2 marks)

e) Briefly describe the "Searle's Chinese room thought experiment" and outline how it could pass the Turing test. (2 marks)

Searle concluded that programs/computers/algorithms are neither constitutive of nor sufficient for minds:

- *programs don't have semantics, programs only have syntax, syntax is insufficient for semantics, every mind has semantics, therefore no programs are minds*
- *minds have intentions and "causal powers" which capture the probability of cause and effect, programs only run formal programs and do not have intentions and "causal powers"*

f) Explain the Virtual mind replies to Searle's conclusions and how they address the concept of the mind. What are Searle's responses to those replies? (2 marks)

Question 6 (6 marks)

A Film Festival has a program of films being screened each marked by a start time (s_i) and finish time (f_i). The problem is to select the maximum number of films that can be viewed by a single person, assuming that a person can only watch one film at a time.

Non-Conflicting films: Let's consider there are N films in a program and for each film, there is a start ***s*** time and finish time ***f*** for that film: **[*s, f*]**. Two films *i* and *j* are said to be non-conflicting if $s_i \geq f_j$ or $s_j \geq f_i$.

Example 1	Example 2	Example 3
Given Program of Films: {[1, 3], [2, 5], [0, 7], [6, 8], [9, 11], [10, 12]} Selected Films: [1, 3] [6, 8] [9, 11]	Given Program of Films: {[1, 3], [2, 5]} Selected Films: [1, 3]	Given Program of Films: {[0, 7], [1, 3], [4, 5]} Selected Films: [1, 3] [4, 5]

Write a greedy algorithm in commented structured pseudocode that tries to select the maximum number of non-conflicting films that can be viewed by a single person at the Film Festival (assuming that the person can only watch one film at a time and stays until the finish time of a film). (6 marks)

Question 7 (13 marks)

a) What is a heuristic? How is it used for problem solving? (2 marks)

b) What is the hill-climbing heuristic? In general terms explain how it works. (2 marks)

c) What are the main features of a hill-climbing heuristic algorithm? (3 marks)

d) What are the problems that can be encountered with hill-climbing heuristic for finding maximums? Explain the possible implications for finding a solution. (2 marks)

e) Explain how the hill-climbing heuristic is used to solve practical problems such as the optimal Travelling Salesman Problem. (2 marks)

f) How would the hill-climbing heuristic need to be altered to solve the decision Travelling Salesman Problem? (2 marks)

Question 8 (10 marks)

a) Complete the missing parts of the pseudocode below for the Floyd Warshall Shortest Path algorithm. (4 marks)

```
Algorithm Floyd-Warshall(G)
# Input graph G a directed or undirected graph of nodes V(G) and edges E(G)
# The graph G may have negative weighted edges, but no negative weight cycles
# initialise an adjacency matrix distance[][] of |V| rows by |V| columns for graph G
For i = 1 to |V| do
    For j = 1 to |V| do
       If (i equals j) then
           distance[i][i] := 0  # zero distance from node i to itself
       Else If (edge i-j exists) then
           [____________________________]  # edge weight from i-j
       Else
           [____________________________]# no direct edge between node i and node j
For k = 1 to |V| do
    For i =1 to |V| do
        For j = 1 to |V| do
            If distance[i][k] + distance[k][j] < [____________________] then
                 [_____________________]:= distance[i][k] + distance[k][j]
```

b) State the time complexity of the Floyd Warshall Shortest Path algorithm with justification. (2 marks)

c) Give a proof of correctness by Contradiction of the Floyd Warshall Shortest Path algorithm, stating any assumptions made. (4 marks)

END OF TRIAL EXAM 2