Office Use Only

MONASH University

Semester One 2015
Examination Period

Faculty of Information Technology

EXAM CODE: **FIT1029**

TITLE OF PAPER: **ALGORITHMIC PROBLEM SOLVING**

EXAM DURATION: 3 hours writing time

READING TIME: 10 minutes

THIS PAPER IS FOR STUDENTS STUDYING AT:(tick where applicable)

☐ Berwick	☑ Clayton	☑ Malaysia	☐ Off Campus Learning	☐ Open Learning
☐ Caulfield	☐ Gippsland	☐ Peninsula	☐ Enhancement Studies	☐ Sth Africa
☐ Parkville	☐ Other (specify)			

During an exam, you must not have in your possession, a book, notes, paper, electronic device/s, calculator, pencil case, mobile phone, smart watch/device or other material/item which has not been authorised for the exam or specifically permitted as noted below. Any material or item on your desk, chair or person will be deemed to be in your possession. You are reminded that possession of unauthorised materials, or attempting to cheat or cheating in an exam is a discipline offence under Part 7 of the Monash University (Council) Regulations.

No exam paper or other exam materials are to be removed from the room.

AUTHORISED MATERIALS

OPEN BOOK	☐ **YES**	☑ **NO**
CALCULATORS	☐ **YES**	☑ **NO**
SPECIFICALLY PERMITTED ITEMS **if yes, items permitted are:**	☐ **YES**	☑ **NO**

Page	Mark	Page	Mark
3		19	
5		21	
7		23	
9		25	
11		27	
13		29	
15		31	
17		Total	

Candidates must complete this section if required to write answers within this paper

STUDENT ID: __ __ __ __ __ __ __ __ __ DESK NUMBER: __ __ __ __ __

This page is intentionally left blank.
Use if needed but it will not be marked.

Part I – (15 marks)

In this section you are required to complete the following phrases. Your answer should be concise. As a guideline, it should require no more space that the space that is provided.

1. A clique in a graph is...

   _____

   _____

2. A decision problem is...

   _____

   _____

3. Prim's algorithm can be used to find...

   _____

4. Prim's algorithm assumes that the input is a graph that is...

   _____

   _____

5. The class P is the set of...

   _____

   _____

6. The class NP is the set of...

   _____

   _____

7. Binary search assumes that the input is...

   _____

This page is intentionally left blank.
Use if needed but it will not be marked.

8. The Johnson-Trotter algorithm is used to...

_____

9. The Edit-distance problem consists of...

_____

_____

_____

10. Applying the brute-force String Matching algorithm to check if the string *ALGORITHM* is a substring of the string *PROBLEALGOMALGORITHMICSOLVING* results in _____ character comparisons. The output of the algorithm in this case is _____.

11. A perfect binary tree is a binary tree such that...

_____

_____

12. The number of vertices in a perfect binary tree of height k is _____.

13. Euclid's algorithm can be used to find _____

_____

14. A function g(n) is said to be O(f(n)) if there exist constants k and L such that...

_____

_____

15. A bitstring is...

_____

_____

This page is intentionally left blank.
Use if needed but it will not be marked.

PART II

Question 1 (2 + 2 + 2 = 6 marks)

Draw the final tree that results from inserting the elements of the list [6, 5, 2, 1, 4, 3] into:

a) A min-heap:

b) A max-heap:

c) A binary search tree:

This page is intentionally left blank.
Use if needed but it will not be marked.

Question 2 (5 marks)
An algorithm is required to determine how to give a specific amount of change. The input of the algorithm should be the *amount* required (in cents) and a list *L[0..n-1]* containing the coin denominations that are permitted (in cents). The output should be a list containing the coin values, that together, add up to the specified amount. For example, a possible result of coin_change(37, [1,5, 10, 20, 50, 100]) is the list [20, 10, 5, 1, 1]. A possible result of coin_change(50, [1,5, 10, 20, 50, 100]) is the list [50], and a possible result of coin_change(12, [1,5, 10, 20, 50, 100]) is the list [10, 1, 1].

Using pseudocode, write an algorithm to solve this problem using the greedy approach. The signature of the algorithm is given below.

coin_change(amount, L[0..n-1])
Returns a list of coins in L, such that the sum of the values adds up to amount.
Input: The amount required, and the list of permitted coins.
Output: A list whose values are in L, such that the sum of the elements equals amount.
Assumption: L is sorted in ascending order and always includes the item 1.

This page is intentionally left blank.
Use if needed but it will not be marked.

Question 3 (2 + 3 + 2 = 7 marks)

Using pseudocode write an algorithm to compute x to the power of N, where N is a non-negative integer. You are required to write 3 versions of this algorithm using three different approaches, as specified below.

a) An iterative version that does not use the divide and conquer principle.

b) A recursive version implementing the divide and conquer principle.

This page is intentionally left blank.
Use if needed but it will not be marked.

c) A version that uses a stack to replace recursion. Use the operations createStack() to create a stack, pop(stack) to pop the item on top of the stack, and push(item, stack) to push *item* onto *stack*.

This page is intentionally left blank.
Use if needed but it will not be marked.

Question 4 (10 marks)

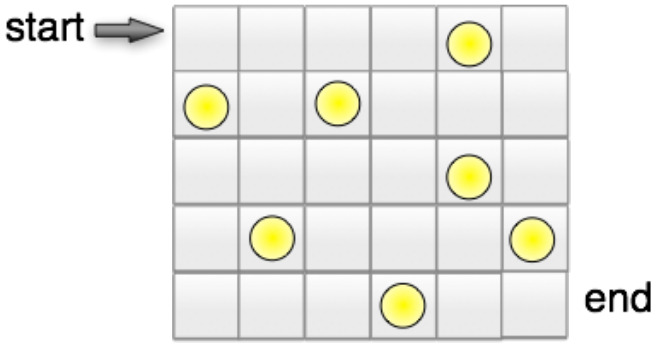
Complete the following table.

Algorithm	Worst case time complexity in Big O notation	Best case time complexity in Big O notation	The worst case happens when…
Sequential search	O(n)	O(1)	The target is not in the list.
Insertion Sort			
Selection Sort			
Quick Sort			
Merge Sort			
Heap Sort			

This page is intentionally left blank.
Use if needed but it will not be marked.

Question 5 (2 + 2 + 6 + 2 = 12 marks)

7 coins are placed on a 5 × 6 board (See figure below). A robot located in the square marked *start* needs to collect as many coins as possible and bring them to the *end* square. At any time, the robot can *only* move one square to the right or one square down of its current location, always picking up any coins found in a square it visits. Diagonal moves, as well as moving up or moving left are not permitted.



a) What is the maximum number of coins that the robot can collect in the board above?

b) The board above can be represented by a table with 5 rows and 6 columns. Complete the sequence below, which gives the optimal path to be followed by the robot. The starting and end positions are given:

[0,0] – – [4,5]

This page is intentionally left blank.
Use if needed but it will not be marked.

c) Considering the rules of this robot we need to determine the maximum number of coins that can be collected for any possible board of arbitrary size n x m. The input is given by a table that contains only 0's and 1's. Ones represent coins and zeroes represent empty spaces. Using pseudocode, write an algorithm that uses the dynamic programming approach to solve this problem. The signature of the algorithm is given below:

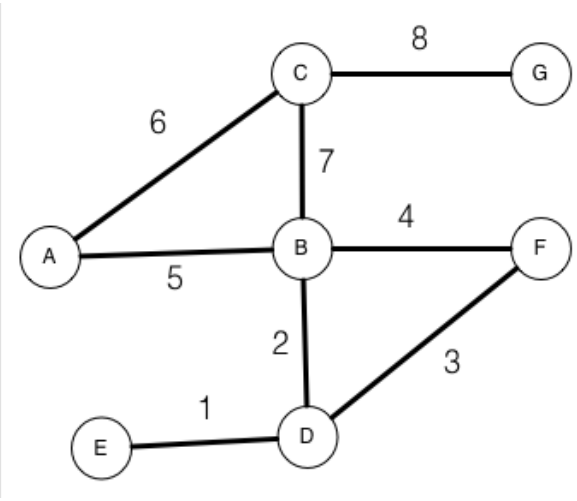
RobotCollectingDynamicProgramming(M[0..n-1, 0.. m-1])
Determines how many coins can be collected from the board represented by table M.
Input: A table M with n rows and m columns, containing only zeroes and ones.
Output: Maximum number of coins that can be collected by the robot

d) Explain how the dynamic programming approach is used to solve this problem. Specify what subproblems are solved at each stage of the procedure.

This page is intentionally left blank.
Use if needed but it will not be marked.

Question 6 (5 marks)

G:



a) The maximum clique size in graph G is _____.

b) The number of cliques with size 3 in G is _____.

c) The total weight of the minimum spanning tree of G is _____.

d) Is graph G Eulerian? _____.

e) Ignoring the edge weights, draw in the space below an adjacency table representing G.

This page is intentionally left blank.
Use if needed but it will not be marked.

Question 7 (5 + 5 = 10 marks)

a) Sort the list [6, 3, 1, 7, 8, 2] in ascending order using selection sort. Report in the table below the state of the list every time an element is swapped.

6	3	1	7	8	2

b) Sort the list [6, 3, 1, 7, 8, 2] in ascending order using insertion sort. Report in the table below the state of the list every time an insertion is performed.

6	3	1	7	8	2

This page is intentionally left blank.
Use if needed but it will not be marked.

Question 8 (9 + 3 = 12 marks)

Partition(L[0..n-1], p, r)
Partitions the List L
Input: A list L and valid indices of that list, p and r such that $0 \leq p < r < n$.
Output: Position of the pivot after the partition.

```
x ← L[p]
i ← p-1
j ← r+1
while(i < j) do {

        do {
          j ← j - 1
        } while(L[j] ≤ x )

        do {
          i ← i + 1
        } while(L[i] ≥ x )

        if (i < j) {
                tmp ← L[i]
                L[i] ← L[j]
                L[j] ← tmp
        }
        //Position A
}
tmp ← L[p]
L[p] ← L[j]
L[j] ← tmp
return j
```

a) Trace this algorithm for input L = [4,1,5,3,7,2,6], p = 0, r = 6. In the table below, report the state of i, j and L every time **//PositionA** is reached.

L	i	j

b) What is the output of Partition([4,1,5,3,7,2,6], 0, 6) and the state of L at the end of the algorithm?

This page is intentionally left blank.
Use if needed but it will not be marked.

Question 9 (3 + 3 = 6 marks)

Showing that a problem is in class P or in class NP requires to give an algorithm related to the problem, and to analyse the time complexity of such algorithm. Using this information show the following. If needed, write an algorithm using pseudocode.

a) Show that the Eulerian path problem “Is there an Eulerian path in a simple and connected graph G?” is in class P.

b) Show that the Hamiltonian path problem, “Is there a Hamiltonian path in a simple and connected graph G” is in class NP.

This page is intentionally left blank.
Use if needed but it will not be marked.

Question 10 (2 + 2+ 3 = 7 marks)

a) Write a module to determine the absolute value of a real number x. The signature is given below:

Abs(x)
Determines the absolute value of x.
Input: A real number x
Output: The absolute value of x.

b) Write a module that given a real number ε, determines if two real numbers x and y are at a distance that is less than ε. You can use the module Abs(x), defined above. The signature is given below:

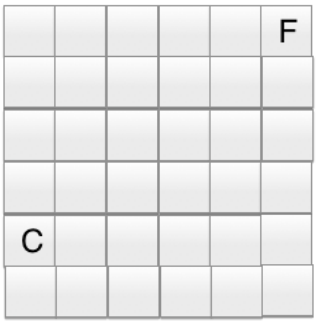
Close(x, y, ε)
Returns True if x and y are at a distance less than ε.
Input: A real number x, a real number y, a real number ε.
Output: True if if the distance between x and y is less than ε, False otherwise.

c) Write a recursive module that given a continuous function f, and an interval [a, b] finds a root of f in the interval [a,b]. That is, finds a number *x*, such that **Abs**(f(x)) < ε. You can use the modules **Abs** and **Close** if necessary. Assume that f(a) and f(b) have different signs.

This page is intentionally left blank.
Use if needed but it will not be marked.

Question 11 (5 marks)

There is a farmer (**F**) and a chicken (**C**) on a 6 x 6 grid. Every turn, both the farmer and the chicken move one square, either horizontally or vertically. No diagonal moves are possible. The farmer is trying to catch the chicken, which happens if she ends a turn on the same square as the chicken. The initial positions of the farmer and the chicken are depicted below.



If the farmer can catch the chicken determine how many moves are required. If the farmer cannot catch the chicken, explain why. Your answer should be concise and fit in the space provided. You do not need to write an algorithm.

This page is intentionally left blank.
Use if needed but it will not be marked.