MONASH University

Office Use Only

Semester One 2014
Examination Period

Faculty of Information Technology

EXAM CODES: **FIT1008**
TITLE OF PAPER: **COMPUTER SCIENCE**
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 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 Monash Statute 4.1.

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	☐ **YES**	☐ **NO**

if yes, items permitted are:

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

STUDENT ID: _ _ _ _ _ _ _ _ _ DESK NUMBER: _ _ _ _ _ _

Page		Mark
3		6
5		8
7		5
9		10
11		10
13		8

Page		Mark
15		8
17		15
19		6
21		6

Total:		**82**

Blank Page

Question 1 (2 marks)

This question is about *sorting*. Describe a list of ***N*** items would be a worst-case input for Bubble Sort, and why?

Question 2 (2 + 2 = 4 marks)

This question is about *lists and complexity*.

Background definition: The *median* of a list is an item x such that half the items are ≤ x and half the items are ≥ x. So the median of the list 3, 4, 7, 7, 9, 18 is 7. (Do not confuse the *median* with the *average*. The two can be very different.)

Consider a *sorted list* of *N* integers. What is the worst-case complexity of finding the median of the list, in each of the following cases, and why? Give the complexities in big-O notation.

a) the list is implemented using an *array*.

b) the list is implemented as a *linked* list.

6

Blank Page

Question 3 (4+2+2 = 8 marks)

This question is about *understanding code and its complexity in BigO*. Consider a stack data type provided by the **`Stack`** class which is implemented using some data structure (you do not need to know which one) and defines the following methods:

```
pop(self)
push(self, item)
__init__(self)
```

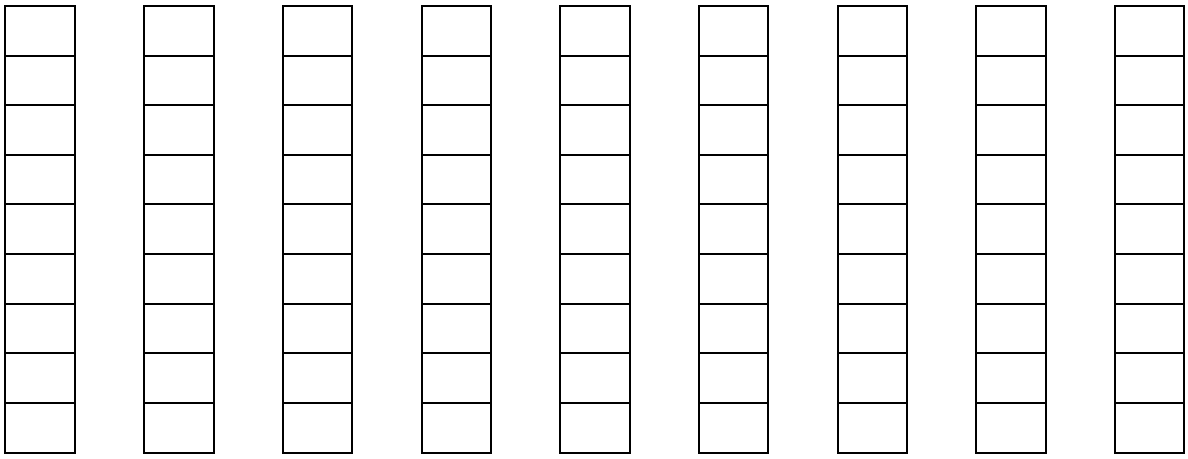
You may assume that the data structure is automatically resizable and therefore *never runs out of space.*

```
def mystery(a_list):
     the_stack = Stack()

     for k in range(len(a_list)):
         the_stack.push(a_list[k])

     for k in range(len(a_list)):
        // HERE
        if (a_list[k] != the_stack.pop()):
            return False
     return True
```

a) Consider the list a`_list = {1,2,3,5,8,6,3,2,1}.` Use the sequence of stacks below to draw the elements on the stack every time the line **`// HERE`** is reached during the execution of **`mystery(a_list)`**



b) Give **a_list** which would cause our **`mystery(a_list)`** method to return **`True`**, and explain why (no explanation means no marks).

c) What is the complexity in Big O of our **`mystery`** method? Explain (no explanation means no marks).

8

Blank Page

Question 4 (5 marks)

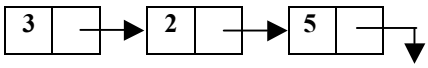
This question is about *basic programming with linked lists*. This question is intended to test your most basic skills at programming with linked lists. Consider a **`Node`** class which defines a node for a linked data structure, and which is defined as follows:

```
class Node:
    def __init__(self, item, link):
         self.item = item
         self.next = link
```

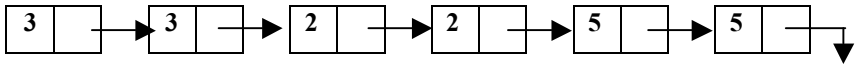
Suppose you have a **`List`** class that implements a Linked List using the Node class above. Add the following method to the **`List`** class:

```
def doubleList(self)
```

which modifies the list pointed to by **`head`** by duplicating every node in the list. For example, consider a **`List`** object **`list`** whose **`list.head`** points to the head of the list:



Then, after calling **`list.doubleList(),list.head`** points to the head of the list:



Important: the only method you can assume you have access to is **`Node(item, link),`** you will need to write any other Python code you need.

5

Blank Page

Question 5 (2 marks)

This question is about *Big O complexity*. Consider an algorithm that performs a certain operation on a list of length **N**. Why is it wrong to consider the case **N = 0** when determining the best case complexity? Explain.

Question 6 (4 marks)

This question is about *choosing data structures*. Several data types (such as stacks, queues, and lists) can be implemented using arrays or using linked nodes. Describe an advantage and a disadvantage of the array implementation compared to a linked node implementation. Explain.

Question 7 (2 + 2 = 4 marks)

Suppose you want to search a list using *binary search*.

a) What data structure would you use, and why?

b) What property must the elements of the list satisfy, and why?

10

Blank Page

Question 8 (5 + 4 + 1 = 10 marks)

This question is about *binary search trees*. Consider the following class for **TreeNode**.

```
class TreeNode:
     def __init__(self, item):
        self.item = item
        self.left = None
        self.right = None
```

a) Suppose you have a **`BinarySearchTree`** class that implements a Binary Search Tree using the **`TreeNode`** class above. Add to **`BinarySearchTree`** class a method **`getMin()`** which returns the *minimum* element in the binary search tree.

b) What is the worst- and best-case complexity of **getMin()**, in big-O notation, in terms of ***N***, the number of items in the tree? Explain.

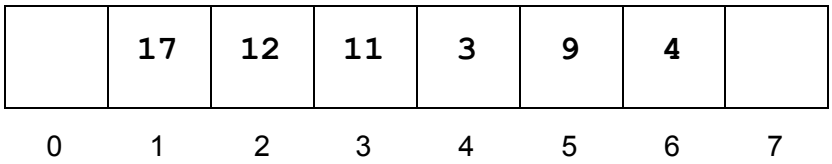
c) What would these complexities be if they were expressed in terms of the *Depth* of the tree, rather than ***N*** ?

10

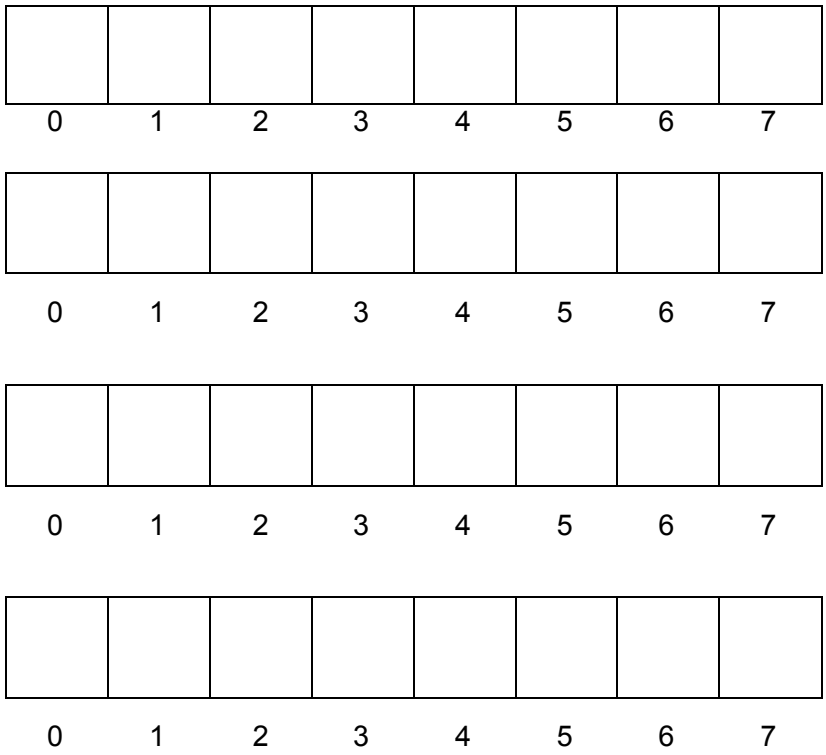
Blank Page

Question 9 (6 + 2 = 8 marks)

A heap is represented using an array. Suppose the contents of the array are as follows.



a) Use the diagrams below to show the contents of the array at each step of the method for getting the maximum element in the heap (**getMax**). You should first show how the array looks just after the maximum element is first replaced. Then show how the array looks after each swap. You may or may not need all the diagrams given.



b) Draw the heap-ordered binary tree that results from applying the **getMax** method to our initial heap.

8

Blank Page

Question 10 (3+3+2 = 8 marks)

This question is about *hash tables*.

Consider a Hash function which returns the following values for the elements

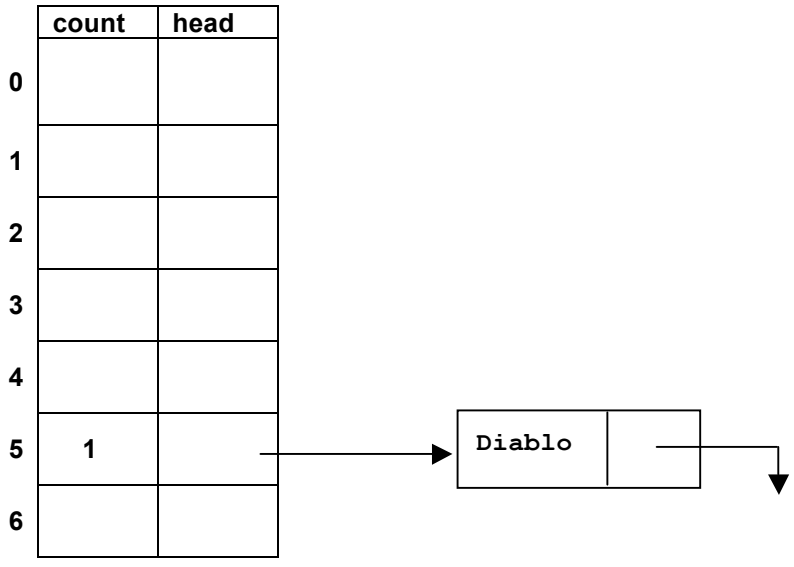
Game	*Hash Value*
StarCr	4
Thief	6
HalfL	1
Splinter	4
DeusEx	1
Doom	4
Diablo	5

a) Using *linear probing*, insert the above games into the following hash table in the order:

StarCr, Thief, HalfL, Splinter, DeusEx, Doom, Diablo

0	**1**	**2**	**3**	**4**	**5**	**6**

b) Complete the following representation of a hash table that uses *chaining*. Insert the names in the same order as in part **a**.



c) Give two characteristics of a good hash function. Explain why they are needed for the function to be good (no explanation means no marks).

8

Question 11 [5 + 10 = 15 marks]

This question is about *MIPS programming and understanding function calls*.

a) Translate the following Python code faithfully into MIPS assembly language. Make sure you follow the MIPS function calling and memory usage conventions.

`def gcd(x, y):`	
`    val = 0`	
`    if (y == 0):`	
`        return x`	
`    val = gcd(y, x % y)`	
`    return val`	

b) Imagine that the **`gcd`** method defined in part (a) of this question is called from a **`main`** method with the line **`gcd(15, 3)`**.

Assume that the **`gcd`** method is located at 0x00400AC, and that at the time the method is run, **`$sp`** is 0x7FFE004C and **`$fp`** is 0x7FFE0050.

Draw what the stack, **`$sp`**, and **`$fp`** will look like:

i. the first time the method is entered, immediately after the declaration of **`val`**;
ii. immediately before the return from the base case.

You do not have to fill in the stack frame for **`main`**.

i.

Name	Value	Address
		0x7FFE0010
		0x7FFE0014
		0x7FFE0018
		0x7FFE001C
		0x7FFE0020
		0x7FFE0024
		0x7FFE0028
		0x7FFE002C
		0x7FFE0030
		0x7FFE0034
		0x7FFE0038
		0x7FFE003C
		0x7FFE0040
		0x7FFE0044
		0x7FFE0048
		0x7FFE004C
		0x7FFE0050

$sp	
$fp	

ii.

Name	Value

$sp	
$fp	

15

Blank Page

Question 12 (4 marks)

Explain what the following fragment of MIPS code does.

```
      addi $s0, $0, 1
      sw $s0, x
ALABEL:    lw $s1, x
      lw $s0, y
      slt $t0, $s1, $s0
      beq $t0, $0, BLABEL
      lw $s0, x
      sll $s0, $s0, 1
      sw $s0, x
      j ALABEL
BLABEL: addi $v0, $0, 1
      add $a0, $s0, $0
      syscall
```

Question 13 (2 marks)

Give two reasons why a programmer might choose to store variables in main memory rather than in registers.

6

Blank Page

Question 14 (2+2=4 marks)

This question is about *binary search trees.* Binary search trees were designed as an alternative to sorted lists.

a) What is the main advantage of a binary search tree over a sorted list implemented using *arrays*? Explain. No explanation means no marks.

b) What is the main advantage of a binary search tree with respect to a sorted list implemented using *linked lists*? Explain. No explanation means no marks.

Question 15 (2 marks)

This question is about *recursive sorting algorithms.* Consider a version of the *quicksort* algorithm in which the pivot is chosen as the last element of the array. Provide an array of integers of length 6 which would have worst case complexity in big O notation for this version of quicksort. Explain why your choice would give the worst case complexity. No explanation means no marks.

End of Exam