Official Use Only

Monash University

Semester One Mid Semester Test 2018

Faculty of Information Technology

EXAM CODES:	**FIT2004 (Mid-semester Test T1)**
TITLE OF PAPER:	**Algorithms and Data Structures**
TEST DURATION:	45 minutes
READING TIME:	5 minutes

THIS PAPER IS FOR STUDENTS STUDYING AT:

☐ Berwick	✓ Clayton	✓ Malaysia	☐ Off Campus Learning	☐ Open Learning
☐ Caulfield	☐ Gippsland	☐ Peninsula	☐ Enhancement Studies	☐ Sth Africa
☐ Pharmacy	☐ 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 in an exam is a discipline offence under Monash Statute 4.1.

No examination papers are to be removed from the room.

<u>AUTHORISED MATERIALS</u>

CALCULATORS	☐ **YES**	✓ **NO**
OPEN BOOK	☐ **YES**	✓ **NO**
SPECIFICALLY PERMITTED ITEMS	☐ **YES**	✓ **NO**

STUDENT ID _ _ _ _ _ _ _ _

Office use only

This page is intentionally left blank. You may write your answers here if more space is needed.

0

INSTRUCTIONS

- You must answer ALL the questions.
- Answers to each question should be in the space DIRECTLY BELOW the questions and (if required) on the blank page overleaf of each question.

General exam technique

Do not throw marks away by **not** attempting all questions. Suppose you get 7/10 on a question for a 20 minutes effort. Spending another half hour on the same question gets *at most* 3 more marks. On the other hand, were you to spend that time on a new question, you might get another 10 marks.

Answer the question that is asked of you. If the question asks for Insertion sort, do not write Quick-sort – this only wastes your time.

Do not write un-necessarily long answers. This wastes your valuable exam time. The question will specifically ask for the information required. Therefore, do not include the information that is not specifically asked for. If asked to justify your answer, provide a clear, logical and concise reasoning.

You do not have to attempt the questions in order. Some questions require less work but may be worth more marks. Carefully read the paper to decide the order in which you should attempt the questions based on the marks associated with each question and whether you know the answer or not.

Best of Luck!

Do not write anything in this table. It is for office use only.

Question	Points	Score
1	8	
2	6	
3	4	
4	6	
Total:	24	

0

This page is intentionally left blank. You may write your answers here if more space is needed.

0

1. This question is composed of short questions. Write your answers to each of these questions in no more than a few lines.

   (a) (2 marks) What is the difference between space complexity and auxiliary space complexity? Give an example of an algorithm for which space complexity and auxiliary space complexity are different.

   Space complexity considers the total space used by the algorithm including the space taken by input. Auxiliary space complexity considers the space taken by the algorithm ignoring the space used by the input. Selection sort has O(N) space complexity and O(1) auxiliary space complexity.

   (b) (2 marks) Write a loop invariant for the following algorithm for sorting an array of positive integers using counting sort. Write a loop invariant that holds at the end of each iteration of the second for loop (write next to #INVARIANT). The loop invariant should able be useful in showing that the algorithm is correct when it terminates. Note that you do NOT need to prove the correctness of the algorithm – just writing a useful loop invariant is sufficient.

```
Find the maximum integer in the array and call it max.
Create an empty array "Count" of size max each value initialized to 0
# Count the number of occurrences of each value
For each value in Input array:
    Count[value]+= 1

Output = empty
For x in (len(Count)):
    n = Count[x]
    Append x to the Output array n times

    #INVARIANT:
```

   #INVARIANT: Output contains all values in Input array that are smaller than or equal to x in sorted order.

4

This page is intentionally left blank. You may write your answers here if more space is needed.

0

(c) (2 marks) Consider a hash table that is using separate chaining where a sorted array is used for chaining. What is the worst-case time complexity of inserting an item in this hash table? Give brief reasoning.

The worst-case cost for insertion is $O(N)$ because the cost for insertion in a sorted array is $O(N)$ in the worst-case.

(d) (2 marks) Radix sort uses a **stable** sorting algorithm to sort each column (as shown in class). Using a small example, illustrate why an **unstable** sorting algorithm cannot be used to sort each column?

Assume two strings `AZ` and `AA`. After sorting on last column, we have `{AA,AZ}`. If we use unstable sorting on the first column, we may get `{AZ,AA}` which is wrong.

4

This page is intentionally left blank. You may write your answers here if more space is needed.

0

2. (a) (3 marks) Assume that you already have an algorithm for partitioning (as shown in class) which partitions the array using a pivot p such that all numbers smaller or equal to p are on its left in the array and all numbers greater than p are on its right. Write **pseudocode** for Quick Select algorithm that returns k-th smallest number in an unsorted array. In your pseudocode, you can refer to the partition algorithm as needed.

```
QuickSelect(k,array):
    choose a pivot p
    partition the array using p
    if index(p) == k:
        return p
    if k < index(p):
        QuickSelect(k,left_array)
    if k > index(p):
        QuickSelect(k,right_array)
```

(b) (3 marks) What is the average-case time complexity of Quick Select? Briefly justify your answer.

Suppose the array is divided into three parts where middle 50% is referred as green. The probability of pivot to be in green is 0.5. When pivot is in the green subarray, the worst-case partitioning is $N/4$ and $3N/4$. Assuming pivot is always in green subarray, the worst-case happens when k-th smallest item is in the bigger partition. So, the worst-case cost when pivot is always in green is $O(N + 3N/49N/16 + ...) = O(N)$. The cost when pivot has 0.5 probability to be in green, is a constant times of this. Thus, it is also $O(N)>$

6

This page is intentionally left blank. You may write your answers here if more space is needed.

0

3. (a) (3 marks) Consider the Python function shown below. Write its recurrence relation for the time complexity and solve it. Also, write its time complexity based on the solution to recurrence relation.

```
def mystery(n):
    if n<=1:
        return 1
    else:
        return n + mystery(n//5)
```

$T(1) = b$
$T(n) = T(n/5) + a$
$T(n/5) = T(n/5^2) + a;$
So $T(n) = T(n/5^2) + 2a.$
$T(n/5^2) = T(n/5^3) + a.$
So, $T(n) = T(n/5^3) + 3a.$
$T(n) = T(n/5^k) + ka.$
Setting $k = log_5(n)$
$T(n) = T(1) + a.log_5(n) = b + a.log_5(n).$
Thus, the time complexity is $O(log(n))$.

(b) (1 mark) What is the space complexity of the `mystery(n)` function. Give a brief reasoning.

The space complexity is $O(log(n))$ which is the space taken by recursion calls.

4

This page is intentionally left blank. You may write your answers here if more space is needed.

0

4. (a) (4 marks) Draw suffix tree for the string `MALALA`.

   ☐

   (b) (2 marks) Give reasoning of why a suffix tree has O(N) space complexity.

The number of leaf nodes is N because there are N suffixes. Each non-leaf node in the tree has at least two children. Thus, the maximum number of parents of the leaf nodes is $N/2$. Similarly, the maximum number of their parents is $N/4$. Therefore, maximum number of nodes is $N + N/2 + N/4 + N/8 + \cdots + 1$ which is $O(N)$.

6

This is the end of the test.

0