Official Use Only

Monash University

Semester Two Mid Semester Test 2017

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.

AUTHORISED MATERIALS

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

STUDENT ID __ __ __ __ __ __ __ __

Office use only

This page intentionally left blank, use if needed but it will not be marked.

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.
- Script book may be used if ADDITIONAL SPACE is required for answering these questions

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	10	
2	3	
3	4	
4	4	
5	4	
Total:	25	

This page intentionally left blank, use if needed but it will not be marked.

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 an in-place algorithm? Is the recursive Quick Sort algorithm an in-place algorithm? Provide brief reasoning for your answer.

   An algorithm that has $O(1)$ auxiliary space complexity is called in-place. Quick Sort is not in-place because the space required for recursive stack is not constant.

   (b) (2 marks) Write a loop invariant for the following Binary Search algorithm. You only need to write the loop invariant that holds at the start of the while loop (write in the given white space).

```
lo = 1
hi = N + 1
while ( lo < hi - 1 )
    # Write loop invariant below

    mid = floor( (lo+hi)/2 )
    if key >= array[mid]
        lo=mid
    else
        hi=mid
if N > 0 and array[lo] == key
    print(key found at index lo)
else
    print(key not found)
```

   key is in $array[1 \cdots N]$ iff key in $array[lo \cdots hi - 1]$

This page intentionally left blank, use if needed but it will not be marked.

0

(c) (2 marks) What is the worst-case time complexity for searching in a hash table using linear probing and how does it compare with the worst-case time complexity for searching using Cuckoo hashing? Give reasoning.

The worst-case cost for search in linear probing is $O(N)$ which is the case when all elements are to be probed to search the key. For Cuckoo hashing, the worst-case cost is $O(1)$ because an element if present must be at its hashed index in T1 or at its hashed index in T2.

(d) (2 marks) Is insertion sort a stable sort? Why or why not?

Insertion sort is stable. This is because swapping stops as soon as the left element is smaller or **equal** to the element.

This page intentionally left blank, use if needed but it will not be marked.

0

(e) (2 marks) Show how the following AVL tree is balanced after 30 is deleted. You need to identify the case (e.g., left-left case) and show how each rotation is done.

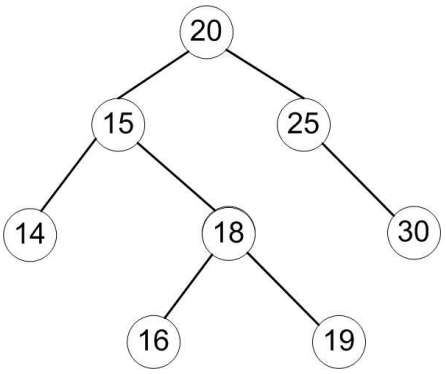
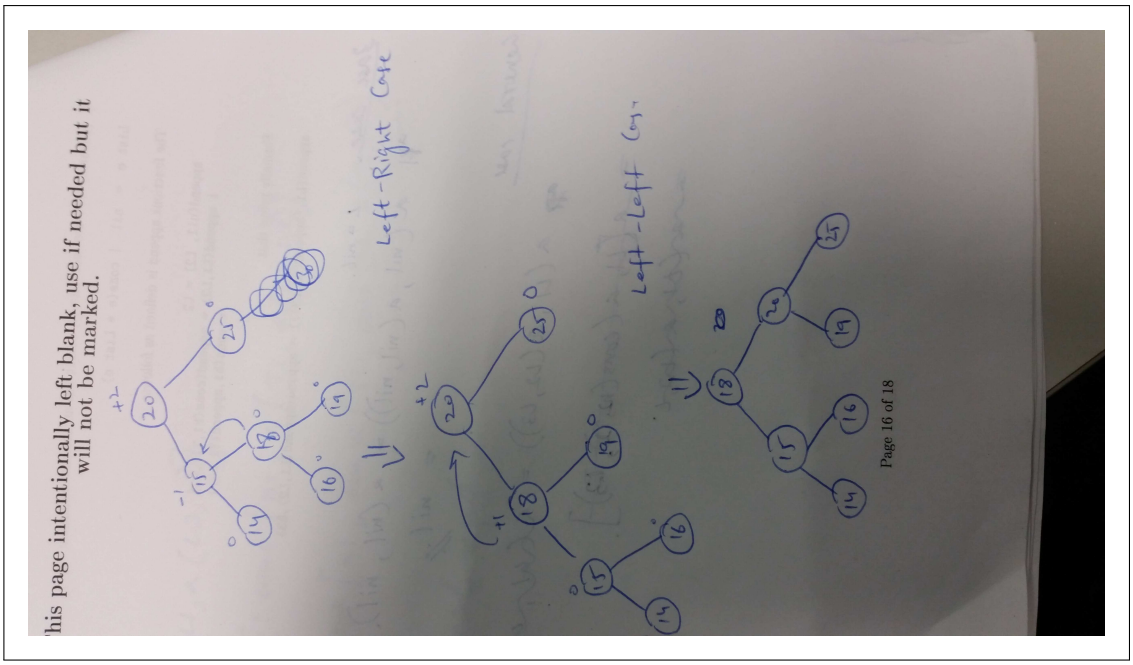


Figure 1: AVL Tree

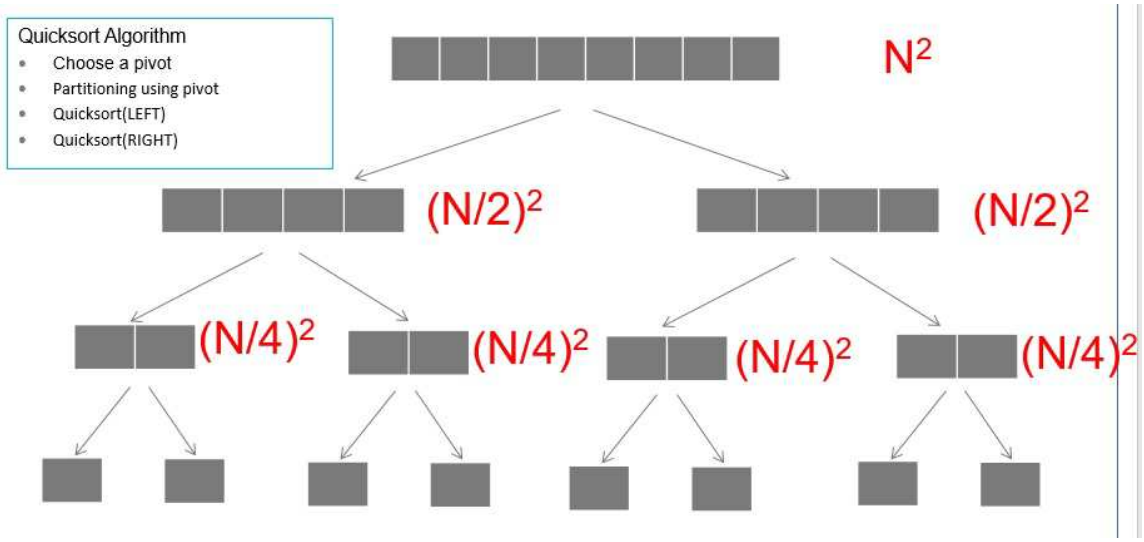


2

This page intentionally left blank, use if needed but it will not be marked.

0

2. (3 marks) Alice has implemented the Quick Select algorithm to find median of numbers in any array. She has then implemented a Quick Sort algorithm where she always chooses median as the pivot and median is computed by her Quick Select algorithm. What is the worst-case time complexity of her algorithm and why?



The worst-case cost for quick select is $O(N^2)$. Hence, the total cost at level 0 is roughly N^2. The total cost for level 1 is $2(N/2)^2 = N^2/2$. The total cost for level 2 is $4(N/4)^2 = N^2/4$. The total cost for level k is $N^2/2^k$.

The total cost for the algorithm is then $N^2(1 + 1/2 + 1/4 + 1/8 + \cdots) = O(N^2)$.

3

This page intentionally left blank, use if needed but it will not be marked.

0

3. (4 marks) Using **induction**, prove that the following **recurrence relationship**:

$$T(N) = \begin{cases} T(N/3) + a, & \text{if } N = 3^k \text{ where } k > 0. \\ b & \text{if } N = 1 \end{cases}$$

has the solution

$$T(N) = b + a \log_3 N$$

where a and b are constants. You **must** use **induction** in your proof.

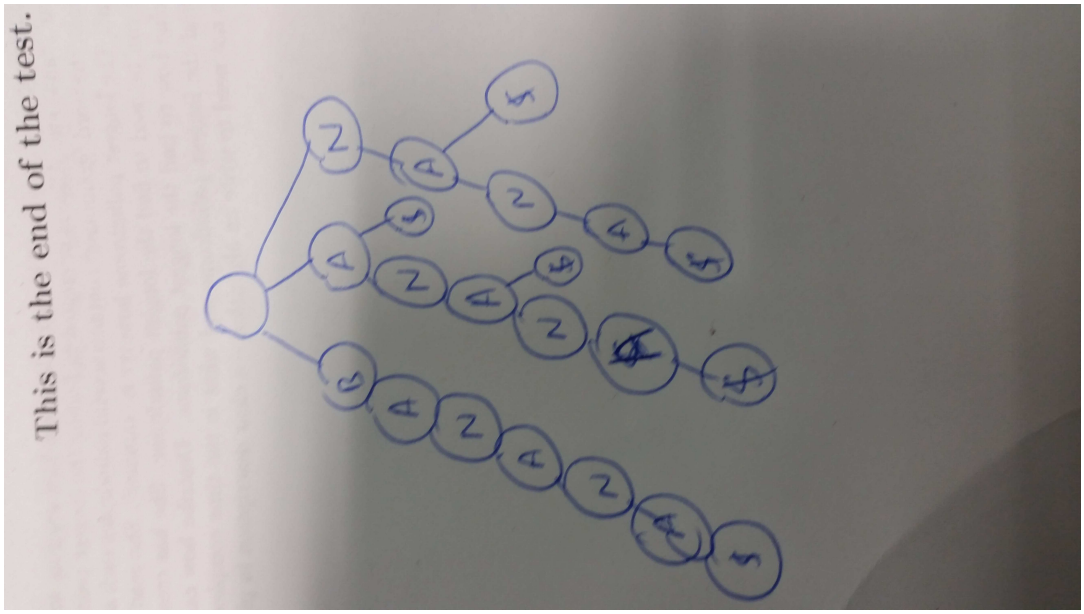
Base case:. $T(1) = b + a \log_3 1 = b$.
Inductive step: Assume that $T(N/3) = b + a \log_3 N/3$.
$T(N) = T(N/3) + a = b + a \log_3 N/3 + a$.
$T(N) = b + a \log_3 N/3 + a \log_3 3 = b + a(\log_3 N/3 + \log 3)$
$T(N) = b + a \log_3 3N/3 = b + a \log_3 N$.

4

This page intentionally left blank, use if needed but it will not be marked.

0

4. (a) (2 marks) Draw a suffix trie of the string "BANANA$".



(b) (2 marks) How can the above suffix trie be used to find the longest repeated substring?

Find the deepest node in the tree containing at least two children, e.g., in the above suffix trie, ANA is the longest repeated substring and the deepest node having two children is A (see the middle A in the suffix ANANA$).

4

This page intentionally left blank, use if needed but it will not be marked.

0

5. (4 marks) Write a **recursive** dynamic programming algorithm for computing N-th fibonacci number. The time and space complexity of your algorithm must be $O(N)$.

```
memo[0] = 0  // 0th Fibonacci number
memo[1] = 1  // 1st Fibonacci number
for i=2 to i=N:
    memo[N] = -1

def fibDP(n):
    if memo[N] != -1
        return memo[N]
    else
        memo[N] = fibDP(N-1) + fibDP(N-2);
        return memo[N]
```

This is the end of the test.

0