Official Use Only

Monash University

Semester Two Mid Semester Test 2017

Faculty of Information Technology

EXAM CODES: **FIT2004 (Mid-semester Test T3)**
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	4	
3	4	
4	3	
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) Radix sort uses a sorting algorithm as a sub-module. Specifically, to sort an array of strings containing M letters each, radix sort calls a sorting algorithm M times. Can selection sort be used as a sub-module of the radix sort? Why or why not?

   No, selection sort cannot be used because it is not stable.

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

```
min = array[1] #note: we assume index starts from 1
index = 2
while index <= N
    # Write loop invariant below




    if array[index] < min
        min = array[index]
    index = index + 1
return min
```

   min corresponds to the smallest element in $array[1...index-1]$.

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

0

(c) (2 marks) What is an in-place algorithm? Give examples of an algorithm that is not in-place and an algorithm that is in-place.

An algorithm that has $O(1)$ auxiliary space complexity is called an in-place algorithm. Insertion sort is in-place whereas quicksort is not in-place.

(d) (2 marks) Write the recurrence relation for the worst-case cost of quick sort.

One of the two below:
T(N) = c*N + T(N-1)
T(N) = N+1 + T(N-1) // assuming partitioning takes N+1 steps

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 37 is added. You need to identify the case (e.g., left-left case) and show how each rotation is done.

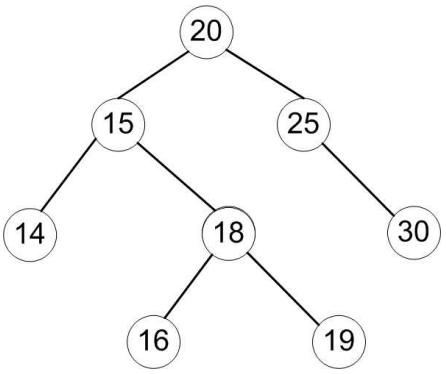
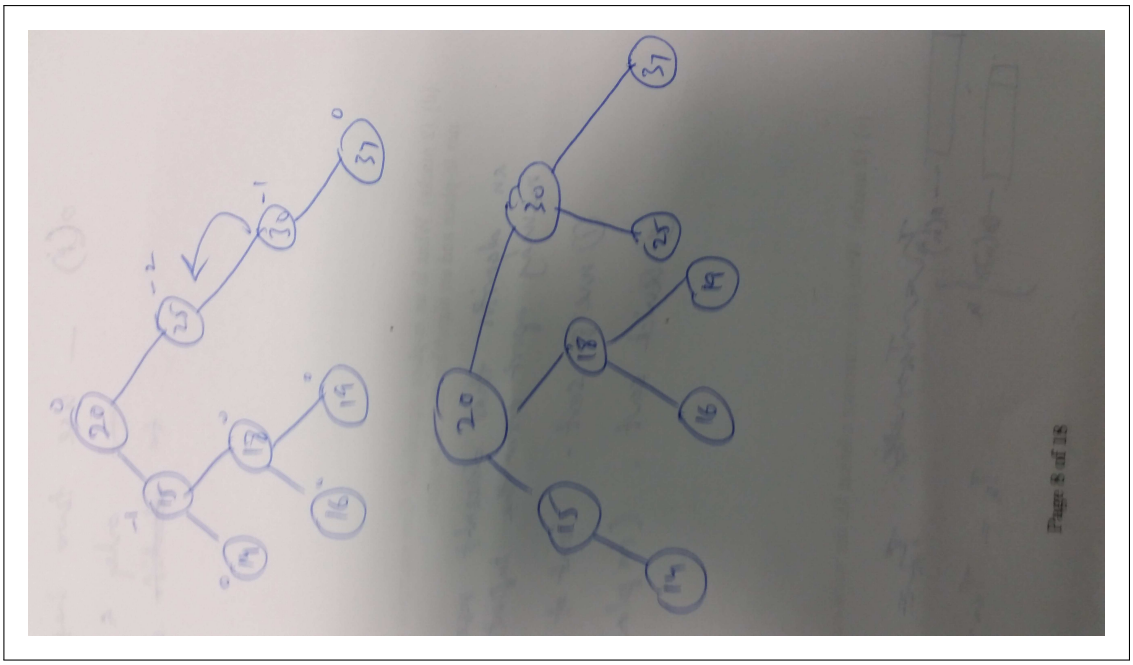


Figure 1: AVL Tree



2

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

0

2. (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]
```


4

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

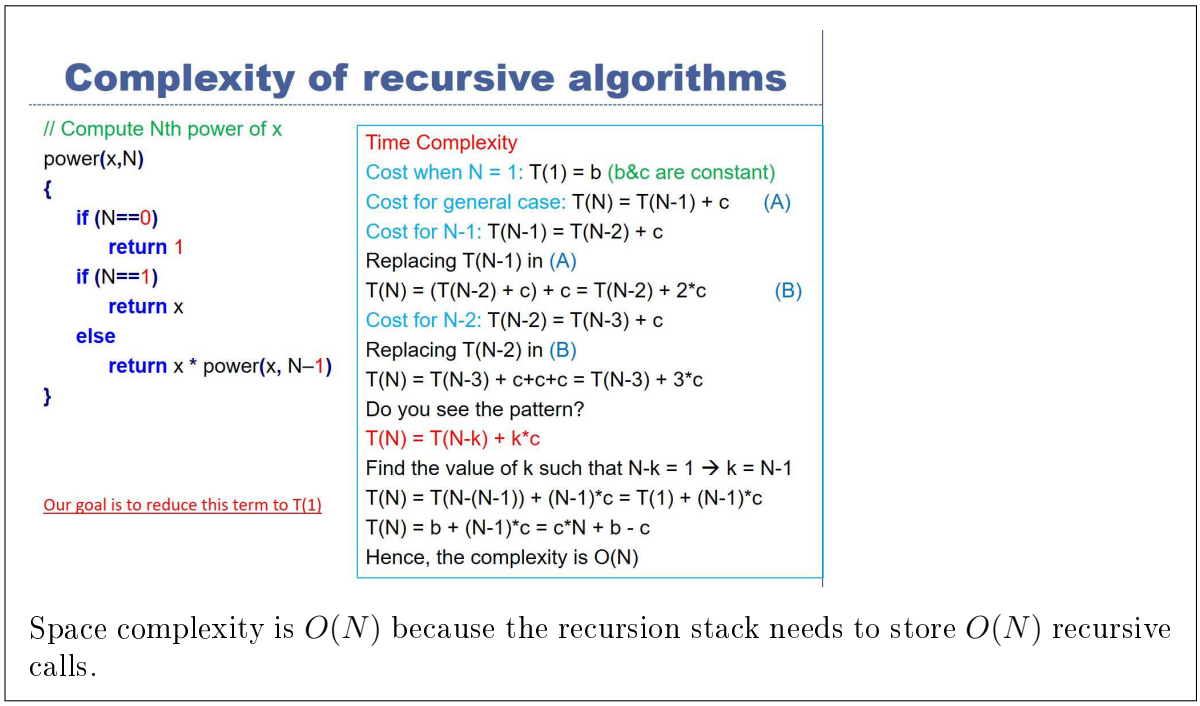
0

3. Consider the following function that returns Nth power of a positive number x where $N \geq 0$.

```
def power(x,N):
    if N == 0:
        return 1
    if N == 1:
        return x
    else:
        return x * power(x,N-1)
```

(a) (3 marks) Write the recurrence relation for the `power(x,N)` function and solve it. What is its time complexity in Big-O notation?

(b) (1 mark) What is the space complexity of `power(x,N)` function? Give a brief justification of your answer.



Space complexity is $O(N)$ because the recursion stack needs to store $O(N)$ recursive calls.

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

0

4. (3 marks) The abstract data type `List` has an abstract definition:

```
List e  =  nil  | cons(e * List e)
```

The functions `append` is defined as follows:

```
append(nil, L2) = L2
    | append(L1,L2) = append(cons(h1,T1),L2)
                    = cons(h1,append(T1,L2))
```

Formally prove that

```
append(L1,(append(L2,L3))) = append(append(L1,L2),L3)
```

See solution in week 1 lecture.

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

0

5. (a) (2 marks) Show the suffix array for the string "BANANA$".

```
Sorted Suffixes:
7 $
6 A$
4 ANA$
2 ANANA$
1 BANANA$
5 NA$
3 NANA$

Hence, suffix array is [7,6,4,2,1,5,3]
```

(b) (2 marks) What is the construction cost (time and space complexity) for a suffix array if merge sort is used for sorting all the suffixes? Give brief reasoning.

Time complexity: $O(N^2 \log N)$. This is because there are $O(N \log N)$ comparisons and each comparison takes $O(N)$.

Space complexity: $O(N^2)$ because N suffixes are to be stored.

This is the end of the test.

0