



The University of Fiji



Undergraduate Programme

Department of Computer Science and Information Technology
Semester 1, Year 2017

FINAL COURSE EXAMINATION PAPER

ITC215: DATA STRUCTURES AND ALGORITHMS

Important Instructions to Students:

1. Read the questions, instructions carefully and answer the questions.
2. You are allowed an **extra 10 minutes reading time** (during this time, you are not allowed to write with pen) and **180 minutes (3 Hours) to complete this paper**.
3. Silent, non-programmable calculators are allowed, but not supplied.
4. Write all your answers in the answer booklet provided. If you need extra sheets, please ask from the supervisor. Write your name, course code and student identification number in the answer booklet and in the attendance sheet.
5. **This Final Course Examination Paper is worth 100 marks contributes 50% of the total course assessments.**
6. This paper contains one (1) section, section A. All questions are compulsory to answer.
7. Write all your answers legibly. The marker has the right to award zero for answers that are not legible.
8. This is a **Closed Book Examination**. The University views plagiarism, cheating and dishonest practice as serious offences and if found, offenders are penalized.

Examination Paper Content (Format):

<u>Section</u>	<u>Question</u>	<u>Content</u>	<u>Marks</u>	<u>Suggested Time</u>
A	1-12	Short Answer Questions	100	180 minutes
		TOTAL	100	180 Minutes

SECTION A:**100 MARKS***All questions in this section are compulsory***QUESTION 1: JAVA PREMIER****[10 Marks]**

- A. What is stored in *ans* as a result of the arithmetic expression, given the following declarations?

```
int ans = 10, v1 = 6, v2 = 8, v3 = 19;  
ans += v1 + 10 - (v2-- / 5) + v3 / v2;
```

[1 Mark]

- B. What is the exact output for *total* after the following segment of code executes?

```
int num = 10, total = 0;  
switch (num)  
{  
  case 1:  
  case 2: total = 5; break;  
  case 3: total = 10; break;  
  case 4: total = total + 3; break;  
  case 8: total = total + 6; break;  
  default: total = total + 4; break;  
}  
System.out.println("The value of total is " + total);
```

[1 Mark]

- C. List and briefly describe the three (3) access control modifiers in Java.

[1.5 Marks]

- D. What is wrong with the following program statement? How can it be fixed?

```
Line 1: System.out.println ("To be or not to be, that  
Line 2: is the question.");
```

[1 Mark]

- E. What output is produced by the following code fragment?

```
String m1, m2, m3;  
m1 = "This is ITC251 FINAL Examination";  
m2 = m1.toLowerCase();  
m3 = m1 + " " + m2;  
System.out.println (m3.replace('h', 'k'));
```

[1.5 Marks]

- F. Write a short Java function, **isMultiple**, that takes two (2) **long** values, **j** and **k**, and returns true if and only if **j** is a multiple of **k**, that is, **j=ki** for some integer **i**.

[4 Marks]

QUESTION 2: OBJECT ORIENTED DESIGN IN JAVA

[10 Marks]

- A. List and describe the three (3) design goals of object oriented programming. [3 Marks]
- B. Write a public static method **threeEqual()** that takes three (3) **integer** values as arguments and returns true if all three numbers are equal, false otherwise. [3 Marks]
- C. Give the type and value of each of the following Java expressions. If an expression will not compile, write **Illegal** under type and put an **X** in value. You must fill in every entry. *Entries left blank will be marked incorrect.* Draw the table in the answer booklet and just provide the number for each expression.

Expression	Type	Value
1. (9 / 2) * 2.0		
2. (9 / 2.0) * 2		
3. !(!!false&& !true)		
4. "2.1" + "3.2"		
5. (1 < 2 < 3)		
6. (1!=2)		
7. 1 + 2.2 + "1.1" + "a"		
8. Integer.parseInt("134")		

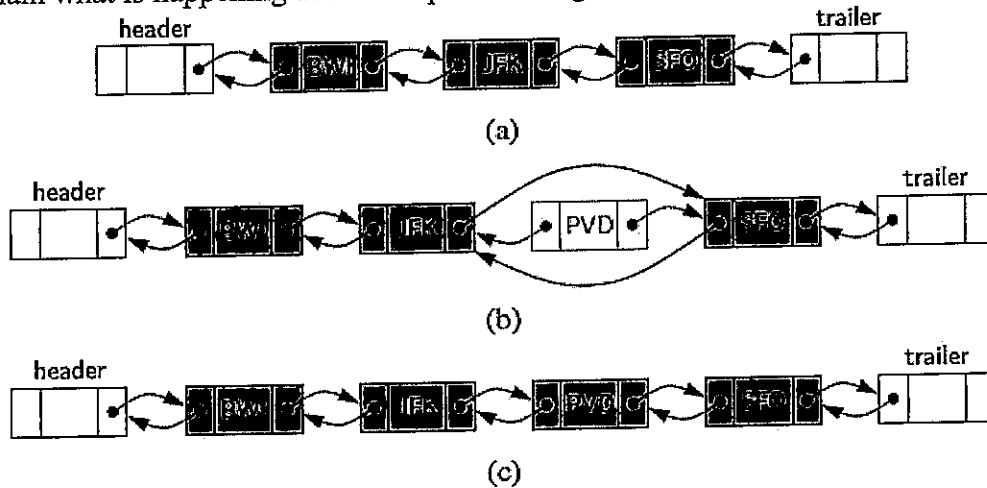
[4 Marks]

QUESTION 3: FUNDAMENTALS OF DATA STRUCTURES

[10 Marks]

- A. Among the following code fragments, write the letters for statements that will not cause a compile-time error. [2 Marks]
- I. `int[] a = int[20];`
 - II. `int[] a = new int[20];`
 - III. `int[10] a = new int[20];`
 - IV. `int[] a = {1, 2, 3}; int b = a;`
 - V. `int[] a;`
 - VI. `int a = {1, 2, 3};`
 - VII. `int[][] a = {{1, 2, 3}, {1, 2, 3}};`
 - VIII. `int[] a = {1, 2, 3}; int[] b = a;`
- B. Explain the following with examples: [3 Marks]
- I. Singly linked list
 - II. Doubly linked list
 - III. Circularly linked list.
- C. Explain with a diagram the *insertion* of an element at the *head* of a *singly linked list*. [2 Marks]

D. Explain what is happening at each step in the diagram below:



[3 Marks]

QUESTION 4: ALGORITHM ANALYSIS

[10 Marks]

A. Order the following functions by asymptotic growth rate. Explain your answer.

$4n \log n + 2n$ 2^{10} $2^{\log n}$ $3n + 100 \log n$ $4n2^n$ $n^2 + 10n$ n^3 $n \log n$

[4 Marks]

B. Compute the complexity of the following code fragment using the **Big O** notation. Explain your answer.

```
public static void twoForLoopsNestedAndNonNested(int n)
{
    for(int i = 0; i < n; i++)
    {
        for(int j = 0; j < n; j++)
        { System.out.println("Printing: " + (i*j));}
    }

    for(int i = 0; i < n; i++)
    {
        System.out.println("Printing: " + i);
    }
}
```

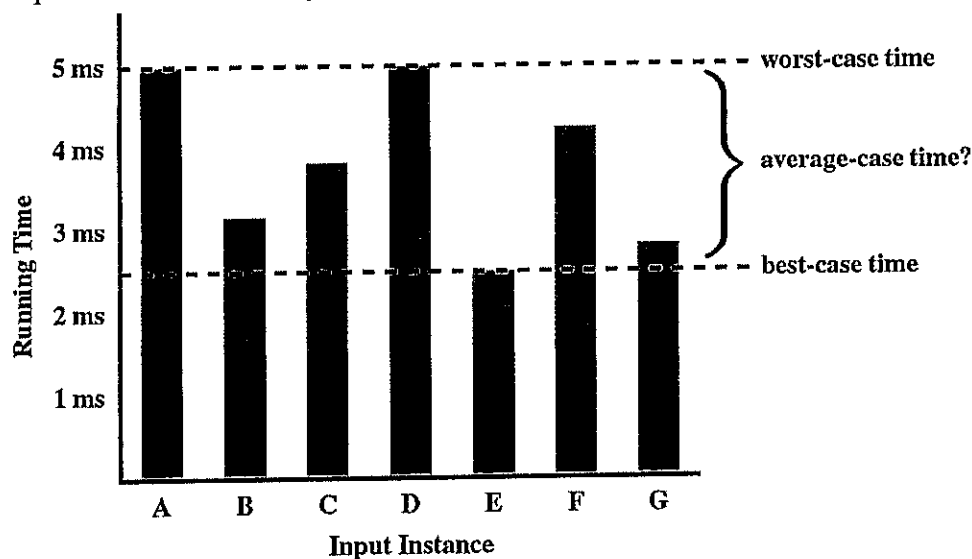
[2 Marks]

C. Compute the complexity of the following code fragment using the **Big O** notation. Explain your answer.

```
public static void twoForLoops(int n)
{
    for(int i = 0; i < n; i++)
    {
        for(int j = n; j > n/2; j++)
        {
            System.out.println("Printing: " + i + j);
        }
    }
}
```

[2 Marks]

D. Explain *worst-case* analysis using the diagram given below:



[2 Marks]

QUESTION 5: RECURSION

[10 Marks]

A. What is Recursion in Java?

[1 Mark]

B. We have learnt three (3) examples of Recursion which are:-*The Factorial method*; *Drawing the English Ruler* and; *Analysing the disk space usage of a file system*. Classify each example as either of type *Linear Recursion*, *Binary Recursion* or *Multiple Recursion* and use the example to explain the three (3) types of recursion.

[3 Marks]

C. What is the output of this program?

```
class recursion
{
    int func (int n)
    {
        int result;
        if (n == 1)
            return 1;
        result = func (n - 1);
        return result;
    }
}
class Output
{
    public static void main(String args[])
    {
        recursion obj = new recursion();
        System.out.print(obj.func(5));
    }
}
```

[1 Mark]

D. Draw the recursion trace for the call factorial five (!5).

[2 Marks]

E. Give a *recursive algorithm* to compute the product of two positive integers, *m* and *n*, using *only* addition and subtraction.

[3 Marks]

QUESTION 6: STACKS, QUEUES AND DEQUES

[10 Marks]

A. Suppose an initially empty stack S has performed a total of 23 *push* operations, 12 *top* operations, and 11 *pop* operations, 4 of which returned null to indicate an empty stack. What is the current size of S?

[1 mark]

B. What values are returned during the following series of stack operations, if executed upon an initially empty stack? push(5), push(3), pop(), push(2), push(8), pop(), pop(), push(9), push(1), pop(), push(7), push(6), pop(), pop(), push(4), push(12), pop(). Answer in a table as below.

Method	Return Value	Stack Content
1. push(5)		
2. push(3)		
3. pop()		
4. push(2)		
5. push(8)		
6. pop()		
7. pop()		
8. push(9)		
9. push(1)		
10. pop()		
11. push(7)		
12. push(6)		
13. pop()		
14. pop()		
15. push(4)		
16. pop(12)		
17. pop()		

[3 marks]

C. Suppose an initially empty queue Q has performed a total of 35 *enqueue* operations, 10 first operations, and 25 *dequeue* operations, 10 of which returned null to indicate an empty queue. What is the current size of Q?

[1 mark]

- D. What values are returned during the following sequence of queue operations, if executed on an initially empty queue? enqueue(5), enqueue(3), dequeue(), enqueue(2), enqueue(8), dequeue(), dequeue(), enqueue(9), enqueue(1), dequeue(), enqueue(7), enqueue(6), dequeue(), dequeue(), enqueue(4), dequeue(), dequeue(). Answer in a table as below.

Method	Return Value	first $\leftarrow$ Q $\leftarrow$ last
1. enqueue(5)		
2. enqueue(3)		
3. dequeue()		
4. enqueue(2)		
5. enqueue(8)		
6. dequeue()		
7. dequeue()		
8. enqueue(9)		
9. enqueue(1)		
10. dequeue()		
11. enqueue(7)		
12. enqueue(6)		
13. dequeue()		
14. dequeue()		
15. enqueue(4)		
16. dequeue()		
17. dequeue()		

[3 marks]

- E. What values are returned during the following sequence of deque ADT operations, on an initially empty deque? addFirst(3), addLast(8), addLast(9), addFirst(1), last(), isEmpty(), addFirst(2), removeLast(), addLast(7), first(), last(), addLast(4), size(), removeFirst(), removeFirst(). Answer in a table as below.

Method	Return Value	first $\leftarrow$ D $\leftarrow$ last
1. addFirst(3)		
2. addLast(8)		
3. addLast(9)		
4. addFirst(1)		
5. last()		
6. isEmpty()		
7. addFirst(2)		
8. removeLast()		
9. addLast(7)		
10. first()		
11. last()		
12. addLast(4)		
13. size()		
14. removeFirst()		
15. removeFirst()		

[2 marks]

QUESTION 7: LIST AND ITERATOR ADTs

[3 Marks]

- A. What does each removeMin call return within the following sequence of priority queue ADT operations: insert(5, A), insert(4, B), insert(7, I), insert(1, D), removeMin(), insert(3, J), insert(6, L), removeMin(), removeMin(), insert(8, G), removeMin(), insert(2, H), removeMin(), removeMin(). Answer in a table as below.

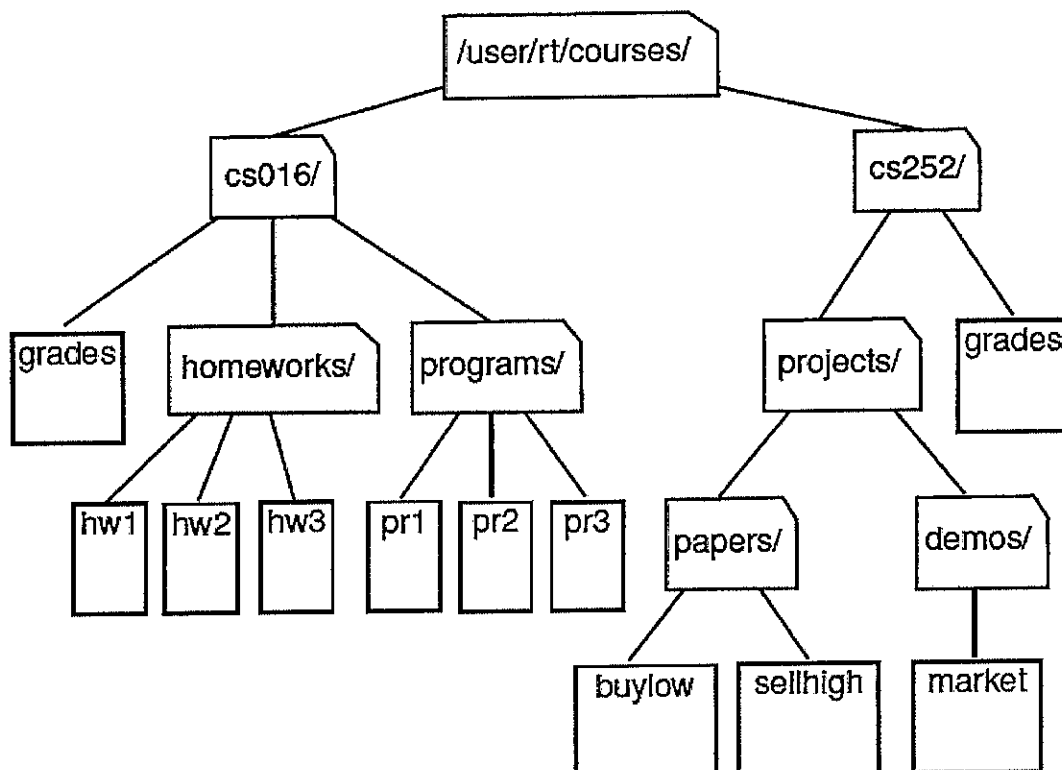
Method	Return Type	Priority Queue Contents
1. insert(5, A)		
2. insert(4, B)		
3. insert(7, I)		
4. insert(1, D)		
5. removeMin()		
6. insert(3, J)		
7. insert(6, L)		
8. removeMin()		
9. removeMin()		
10. insert(8, G)		
11. removeMin()		
12. insert(2, H)		
13. removeMin()		
14. removeMin()		

[3 Marks]

QUESTION 8: TREES

[10 Marks]

- A. Consider the tree given below and answer the questions that follow.



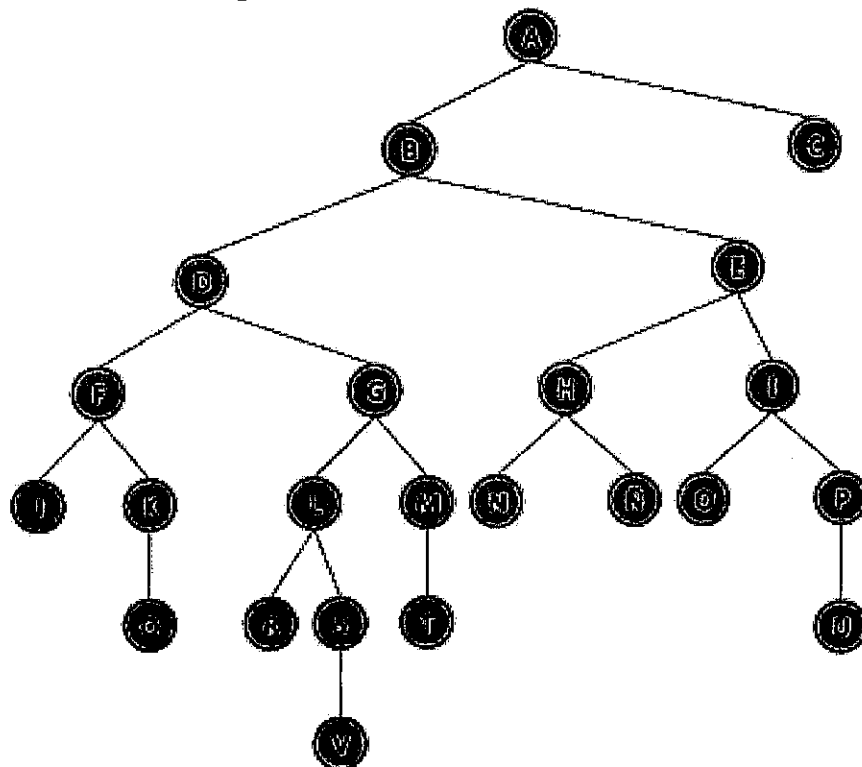
- I. Which node is the *root*?
- II. What are the *internal nodes*?
- III. How many *descendants* does node cs016/ have?
- IV. How many *ancestors* does node cs016/ have?
- V. What are the *siblings* of node homeworks/?
- VI. Which nodes are in the *subtree* rooted at node projects/?
- VII. What is the *depth* of node papers/?
- VIII. What is the *height* of the tree?

[4 Marks]

B. Draw the binary tree representation of the following arithmetic expression:
 $((5 + 2) * (2 - 1)) / ((2 + 9) + ((7 - 2) - 1) * 8)$.

[2 Marks]

C. Given the following tree, write the pre-order, in-order and post-order traversals.



[3 Marks]

D. At which nodes of a *maximum heap* can an entry with the *largest key* be stored?

[1 Mark]

QUESTION 9: PRIORITY QUEUES

[2 Marks]

- A. Illustrate the execution of the *selection-sort* algorithm on the following input sequence: (22,15,36,44,10,3,9,13,29,25). Draw the selection sort at each step.

[2 Marks]

QUESTION 10: MAPS, HASH TABLES AND SKIP LISTS

[10 Marks]

- A. Draw the 11-item hash table that results from using the hash function $h(i) = (2i+5) \bmod 11$ to hash the keys 12, 44, 13, 88, 23, 94, 11, 39, 20, 16 and 5, assuming collisions are handled by *chaining*. Show your calculations.
- B. What is the result of question A assuming collisions are handled by *double hashing*, using the secondary hash function $h'(k) = 7 - (k \bmod 7)$? Show your calculations.
- C. Show the result of question A ; assuming collisions are handled by *quadratic probing*, up to the point where the method fails because no empty slot is found. Show your calculations.
- D. Insert the numbers 5, 2, 1, 6, 8, 3, and 4 into an initially empty *Skip List* - assuming that a count of consecutive heads is used to determine the insertion level, and that the judicious coin has produced the following result: HTTHTHHTTHHHTT ... Explain your answer.

[2 Marks]

[2 Marks]

[2 Marks]

[4 Marks]

QUESTION 11: SEARCH TREES

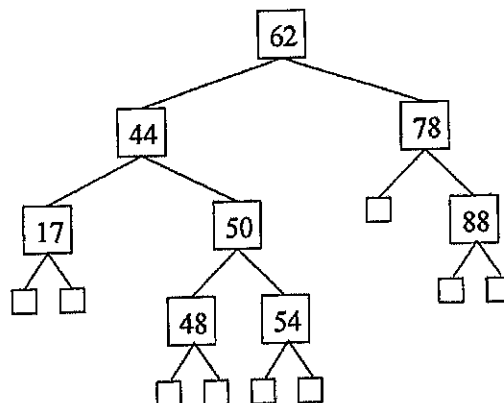
[10 Marks]

- A. If we insert the entries (3, C), (1, A), (5, E), (4, D), and (2, B) in this order, into an initially empty binary search tree (BST), what will it look like?
- B. How many different binary search trees can store the keys {1, 2, 3}? Draw all possible trees.
- C. Consider the following AVL tree and Draw the AVL tree resulting from the removal of the entry with key 62 from the AVL tree.

[2 Marks]

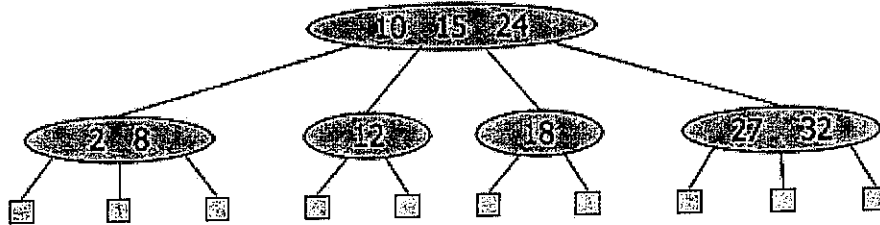
[2 Marks]

[2 Marks]



- D. Consider the set of keys $K = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\}$. Draw a **(2,4) tree** storing K as its keys using the fewest number of nodes. [2 Marks]

- E. Convert the following 2-4 tree into a **red-black tree**. Provide a key to differentiate between red nodes and black nodes.



[2 Marks]

QUESTION 12: SORTING AND SELECTION

[5 Marks]

- A. List and explain the three steps of the **divide-and-conquer** pattern. [3 Marks]
- B. Given the following array as input, illustrate how the **Mergesort algorithm** performs sorting. Draw sort at each step of divide and conquer.

3	8	4	10	1	5	6	9
---	---	---	----	---	---	---	---

[2 Marks]

THE END

