THE UNIVERSITY OF THE SOUTH PACIFIC

CS214: Design & Analysis of Algorithms
STEMP

Special Final Examination
Semester II, 2023

F2F/Blended Mode

Duration of Exam: 3 hours + 10 minutes

Reading Time: 10 minutes

Writing Time: 3 hours

Total Marks: 40

The exam covers the following learning outcomes:

CLO – 1: Evaluate the efficiency of algorithms (32.5%)

CLO – 2: Assess the suitability of different algorithms/data structures for solving a given problem (17.5%)

CLO – 3: Solve computationally difficult real world problems using appropriate algorithmic techniques (50%)

Instructions:

1. Write your answers in the space provided in this Question Paper.
2. Answer all questions. There are 9 questions and all questions are compulsory.
3. This exam is worth 40% of your overall mark. The minimum mark to pass the final exam is 16/40.
4. The total number of pages including this cover sheet is 14.
5. This is a closed book exam. No printed materials and electronic devices are allowed.

ID:_____	**Name:**_____
SeatNo: _____	**Campus:** _____

This page is intentionally left blank

1. Java provides ArrayList and LinkedList data structures for data manipulation. What is your preference and why, when the efficiency of an algorithm is your main concern? Justify your choice clearly. (2+2 = 4 marks)

[8 mins]

2. Determine the time complexity of the following code. Show your working. (3+3 marks)

   a) Find the worst-case time complexity:

```
final int P = 200;
final int Q = 100;//where Q is always less than P

for (int i = 0; i < P; i++) {
    for (int j = 0; j < Math.min(i,Q); j++) {
        System.out.println(j);
    }
}
```

[12 mins]

b) Determine the best time complexity:

```
int a = 0;
for (i = 0; i < N; i++) {
    for (j = N-5; j < N; j++) {
        a = a + i + j;
    }
}
```

10 mins

3. How would you determine the big O order of a given algorithm empirically?

(3 marks)

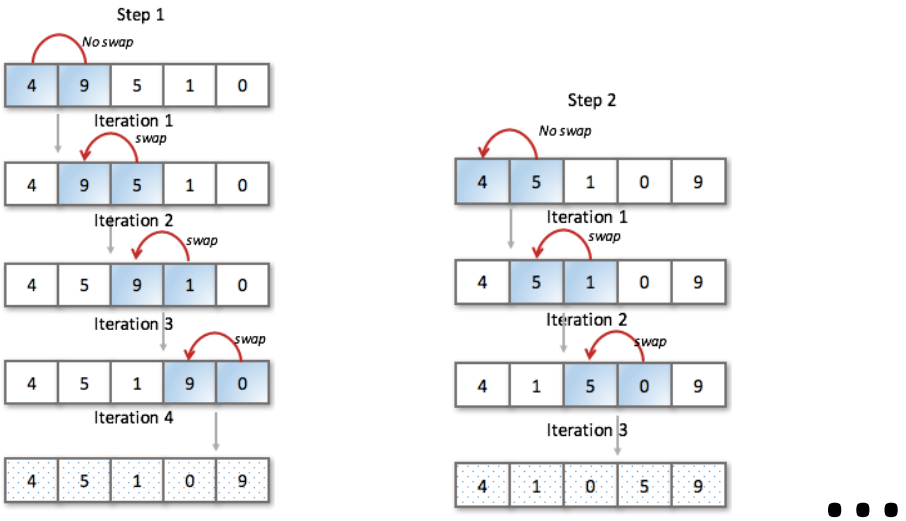
[5 mins]

ID: _____

4. A variation of the *bubble sort* algorithm and an example illustrating what it does are shown below. Using this algorithm answer the following questions:

```
static <T extends Comparable> void bubbleSort (LinkedList<T> S){
    boolean exchanged;
    int n = S.size()+1;
    int counter = 0;
    do {
        n--; //make loop smaller each time
        exchanged = false;
        for ( int i=0; i < n-1; i++ ) {
            if (S.get(i+1).compareTo(S.get(i)) < 0) {
                exchange (S, i, i+1); //exchanges S.get(i) with S.get(i+1)
                exchanged = true; //after exchange must revisit
            }
            counter++;
        }
        System.out.println(S);
    }while (exchanged);

    System.out.println("X(n) = " + counter);
}
```



a. Evaluate the Big O order of the best and worst time complexity for this algorithm. Show all working. (3 marks)

b. Under what condition(s) you will attain the best and worst time complexities? (3 marks)

(answer both questions 4(a) and 4(b) here)

[6 mins] + [6 mins]

5. Threads enable multiple tasks to run simultaneously. You are making a program to run two evolutionary algorithms GA and RA run simultaneously. You want to show the progress of both algorithms on a GUI screen where Y-axis should be the fitness value and X-axis should be the generation count. The following code shows the implementation of the Algorithm class including GUI work. (Note: … means that code is hidden) (4 marks)

```
class Algorithms{

   public static void GA(){…} //… means hidden code
   public static void RA(){…} //… means hidden code
```

[15 mins]

Hint: First create threads for GA and RA and then run them simultaneously. You may also need the following code:

Interface runnable
run(): When an object implementing interface Runnable is used to create a thread, starting the thread causes the object's run method to be called in that separately executing thread.

Class Thread
Thread (Runnable target): Allocates a new Thread object.
start(): Causes this thread to begin execution; the Java Virtual Machine calls the run method of this thread.
Boolean isInterrupted(): Tests whether this thread has been interrupted.

6. What kind of algorithmic design paradigm you will choose for the following large problems that can be practically implemented? Please justify your answer:

Knapsack Problem

The 0-1 knapsack problem is a common optimization problem used in teaching optimization. It's objective is to maximize the profit from a given set of weights and the weight limit. Knapsack is an NP-complete problem.

NQueen Problem

The n-queens problem is about finding how many different ways queens can be placed on a chessboard so that none attack each other. The n-Queen problem becomes intractable for large 'n' values and is thus placed in NP (Non-Deterministic Polynomial) class problem. (4 marks)

[8 mins]

7. Draw the Huffman tree corresponding to the encoding table below. Additionally, find the possible values of I from 1-15. (4 marks)

Character	Frequency	Code
B	2	01111
F	1	01110
H	3	0110
I	10	00
L	?	010
M	15	10
S	15	11

[15 mins]

8. Consider the 0-1 knapsack problem with four different items having different weights and profit values;
   item 1 (3kg, $5)
   item 2 (1kg, $2)
   item 3 (4kg, $7)
   item 4 (2kg, $3)
   item 5 (5kg, $8)

   The knapsack has a capacity of 10 kg.

Do the following by showing your **complete working**:

a. Apply greedy approach to compute the optimal profit in filling the knapsack (3 marks)

[10 mins]

b. Apply a dynamic programming approach to solve the same problem. Show all possible solutions. (3 marks)

[15 mins]

c. Apply the backtracking approach for the above knapsack problem. Show the State Space tree. (3 marks)

[15 mins]

EXTRA SHEET

END