NATIONAL SCHOOL OF BUSINESS MANAGEMENT

BSc Management Information Systems (Special)-20.3
BSc. (Honours) in Software Engineering – 20.3
BSc. (Honours) in Computer Science – 20.3
BSc. (Honours) in Computer Networks – 20.3

Year 01 Semester 02 Examination
13 October 2021
CS106.3 – Algorithms and Data structures

Instructions to Candidates

1) **Answer all questions.**

2) Total Number of Pages five (05).

3) **Time allocated for the examination is five (05) hours (Including downloading and uploading time).** Please type your answer unless a diagram is required. Diagrams can be handwritten and attached as a figure.

4) Weightage of Examination: 60% out of final grade

5) Download the paper, provide answers to the selected questions in a word document.

6) Please upload the document with answers (Answer Script) to the submission link before the submission link expires. Answer script should be uploaded in PDF Format

7) Under any circumstances E-mail submissions would not be taken into consideration for marking. Incomplete attempt would be counted as a MISSED ATTEMPT.

8) The Naming convention of the answer script – Module Code_Subject name_Index No (E.g. CS106_DSA_100065)

9) You must adhere to the online examination guidelines when submitting the answer script to N-Learn.

11) Your answers will be subjected to Turnitin similarity check, hence, direct copying and pasting from internet sources, friend's answers etc. will be penalized.

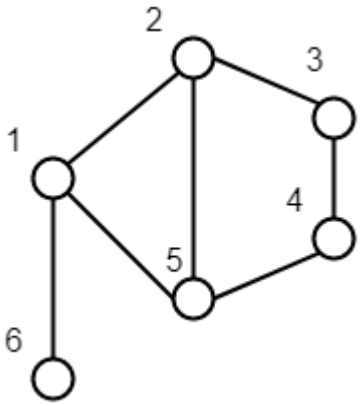
01. The following questions are based on the sequential data structures covered during the module. (Total= 20 Marks)

I. Graphically illustrate the insert (push) and remove (pop) functions. You may insert each integer of your index no into the stack for the illustrations and remove the 4th integer from the stack. Note: You're required to mark how the 'top' position changes during the execution. (4 Marks)

II. Graphically illustrate the insert (enqueue) and remove (dequeue) functions. You may insert each integer of your index no into the queue for the illustrations and remove the 2nd integer from the queue. Note: You're required to mark how the 'front' and 'rear' positions change during the execution. (4 Marks)

III. Write down a code snippet for pop() function and dequeue() function. Put careful attention to the order of steps. (4 Marks)

IV. Derive the DFS and BFS output of the following graph. Identify the suitable data structures used in these two scenarios. Assume it starts from "0" and follows ascending order. (4 Marks)



V. Evaluate the following postfix notations and derive the final output. What is the suitable data structure to be used in this scenario? (4 Marks)

a) 2 3 4 *5 + -

b) 6 2 5 + - 3 8 2 / + *2 ^ 5 +

02. The following questions are based on searching algorithms. **(Total= 20 Marks)**

Consider the following array to answer Part I.

3	39	18	25	60	76	90	9	13

I. Graphically illustrate how to search for value '60' using linear search. (4 Marks)

II. The algorithm below uses a different method to search through the array for a name. Fill in the gaps to complete the algorithm. (4 Marks)

```
array person[5]
people = ["Iman", "Flex", "Kim", "Zoe", "Gavin"]
found = False
x = 0
searchfor = input("enter a name to search for : ")
while found = False ------- x <5
          if people[x] = searchfor then
              found = ------
              print "found at position" + -----
              x = x + 1
          endif
---------
```

III. Fill the following table assuming that you're applying binary search (remind what we did at the class) on the below data set given. Note. Copy the table to your answer sheet. (4 Marks)

array[]=[3,9,13,21,30,39,42,48,52,60,76,79,90] and search key (sk) is 48

Variable	**Initially**	**After Iteration 1**	**After Iteration 2**	**After Iteration 3**
sk	48			
Size of the array	13			
First index	0			
Last Index	12			
found	false			
position	-1			
(!found && first<=last)	true			
middle	n/a			
array[middle]	n/a			

IV. Write down the Big-oh notation of the linear and binary search algorithms. Justify your answers. (4 Marks)

03. The following questions are based on sorting algorithms. (Total= 20 Marks)

Consider the following array to answer Part I and III

3	39	18	25	60	76	90	9	13

I. Graphically illustrate how to apply the bubble sort algorithm to sort the array. (5 marks)

II. Graphically illustrate how to apply the merge sort algorithm to sort the array. (5 marks)

III. Now take your birthdate (e.g. 2018-10-02 as 2,0,1,8,1,0,0,2 so it will look like as below array) as the input. Graphically illustrate how to apply the insertion sort algorithm to sort the integer set. Note: We need to make the integers set in descending order. (6 Marks)

Note: If you have born on 2018-10-02, your input array will look like this:

2	0	1	8	1	0	0	2

IV. If the assignment operation is considered costly, what sorting algorithm would you select as the preferred algorithm to sort your dataset? Justify your answer. (4 Marks)

04. The following questions are based on Binary Trees and related concepts. (Total= 20 Marks)

I. Draw the binary search tree that results from inserting the following sequence into an initially empty tree. (5 Marks)

June, Marie, Billy, Alan, Tob, Fredi, Janifer, Jovee, Davinci

II. Derive the pre-order, post-order and in-order traversal output for the above resultant tree. (6 Marks)

III. Identify the path to node Jovee. What is the depth of this path? (3 Marks)

IV. Consider the following expression 2*((8+1)/6)+(15-5). Now derive the equivalent expression tree. Derive the postfix notation of the expression on the same. (6 Marks)

05. The following questions are based on Complexity Analysis and Recursion. (Total= 20 Marks)

I. Simplify the following expressions (5 marks)

a. $7009 + 23n$
b. $\log n - n^2$
c. $n\log n + 8n^2 + 600n$
d. $n! + 98n^2$
e. $12n! - 7n^n + 2^n$

II. Evaluate the complexity of the following two algorithms and decide on the best algorithm to get the sum of integers. (5 marks)

Algorithm A	**Algorithm B**

int addall(int n){
return n* (n+1)/2
}
	int addall(int n){
	int total;
	for(int i=1; i<=n;i++){
	total+=i;
	}
	return n* (n+1)/2
	}

III. Euclidean algorithm for calculating GCD is defined as follow:

if **num2** == 0 then GCD (**num1**, **num2**) is **num1**

else GCD (**num1**, **num2**) is GCD (**num2**, modulus(**num1** / **num2**))

Refer to the program below implemented for computing factorials and write a recursive code for the Euclidean GCD algorithm. (5 marks)

```
long factorial(int n)
{
  // base case 0! = 1! = 1
        if (n <= 1)
              return 1;
  return(n * factorial(n-1));
}
```

IV. Consider the following series, identify the series's logic, and derive a recursive algorithm. Assume you have input 5 (n=5). Now derive the recursive call tree. (5 marks)

Input (n)	Output
0	0
1	1
2	1
3	2
4	3
5	5
6	8
7	13
8	21
9	34
10	55

……END OF PAPER……