



NATIONAL SCHOOL OF BUSINESS MANAGEMENT

BSc. (Honours) in Software Engineering – 20.1

BSc. (Honours) in Computer Security – 20.1

BSc. (Honours) in Computer Networks – 20.1

Year 01 Semester 02 Examination

12-01-2021

CS105.3 – Data Structures and Algorithms

Instructions to Candidates

- 1) Answer all questions.
- 2) Total Number of Pages four (04).
- 3) Time allocated for the examination is three (03) hours and 30 minutes (Including downloading and uploading time)
- 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
- 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. Compare and contrast an Array vs Linked List and Stack vs Queue. (6 Marks)
- II. Graphically illustrate the insert (push) and remove (pop) functions of a stack. You may insert **each integer in your index no** into the stack for the illustrations and remove the 2nd integer from the stack. Note: You are required to mark how 'top' position change during the execution. (5 Marks)
- III. Graphically illustrate the insert (enqueue) and remove (dequeue) functions of a queue. You may insert **each integer in your index no** into the queue for the illustrations and remove the 2nd integer. Note: You are required to mark how 'front' and 'rear' position change during the execution. (5 Marks)
- IV. Write down a code snippet for *pop()* function and *dequeue()* function. Pay careful attention to the order of steps. (4 Marks)

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

- I. Compare and contrast linear search algorithm and binary search algorithm. Identify their time complexities in Big O notation. (6 Marks)

Consider the following array to answer Part II and Part III

12	10	8	27	18	45	1	90	35
----	----	---	----	----	----	---	----	----

- II. Graphically illustrate how to search for value '35' using linear search. (5 Marks)
- III. Graphically illustrate how to search for value '35' using binary search. (5 Marks)
- IV. If there are 1000 elements in an array, explain the best case, average case, and worst-case scenario in linear and binary search algorithms. (4 Marks)

03. The following questions are based on sorting algorithms.

(Total= 20 Marks)

Consider the following array to answer Part I and III

12	10	8	27	18	45	1	90	35
----	----	---	----	----	----	---	----	----

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

2	0	1	8	1	0	0	2
---	---	---	---	---	---	---	---
- IV. In the best-case scenario of the bubble sort how many times you need to run your algorithm? Identify it's time complexity in Big O notation. Develop a code snippet to illustrate bubble sort algorithm. (4 Marks)

04. The following questions are based on Recursion and Time complexity analysis concepts.

(Total= 20 Marks)

For the below question, we will take your Destiny Number as the input. When calculating the Destiny Number, it reduces the month, day, and year to single digits. It is common for some numbers to take multiple reductions. In such cases, repeat the same technique until a single digit or Master Number is achieved. E.g.

Month: $12 = 1 + 2 = 3$

Day: $17 = 1 + 7 = 8$

Year: $1986 = 1 + 9 + 8 + 6 = 24 = 2 + 4 = 6$

Next take the resulting three single digits or Master Numbers and add them: $3 + 8 + 6 = 17 = 1 + 7 = 8$.

And so, in the instance of December 17, 1986, we get a Destiny Number of 8.

- I. Write a code snippet to derive the recursive Fibonacci function. (4 Marks)
- II. Based on the derived function in (II) now draw the recursive call tree if you input x. (6 Marks)
Note: $x = 5$ if *Destiny Number* > 5 else $x = \text{Destiny Number}$.

III. Derive the output of below code snippets if you pass *Destiny Number as the input*. (6 Marks)

a.

```
void recSolv(int n){
    if (n<0){
        printf("%d", n);
        recSolve(n-1);
    }
}
```

b.

```
void recSolv(int n){
    if (n<0){
        recSolve(n-1);
        printf("%d", n);
    }
}
```

IV. Derive the code complexity of the following source code. Justify your answer. (4 marks)

```
int main(){
    int n = 10;
    int sum = 0;
    for (int i=0; i<n; i++)
        sum+=2;
    cout =sum;
    return 0;
}
```

05. The following questions are based on Binary Trees and related concept. **(Total= 20 Marks)**

- I. Draw the binary search tree that results from inserting the following sequence into an initially empty tree. (5 Marks)
50, 40, 80, 30, 20, 88, 75, 100
- 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 88. What is the depth of this path? (3 Marks)
- IV. What is the total value of leaf nodes? (2 Marks)
- V. If there are 15 nodes in a perfect binary tree. What would be the height? (2 marks)
- VI. Using the resultant tree in "I" explain how to search for node 100. (2 marks)

.....**END OF PAPER**.....