Daffodil International University
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Midterm Exam Examination, Fall 2021 @ DIU Blended Learning Center
Course Code: CSE213 (Eve), Course Title: Algorithm
Level: 1   Term: 1   Section:
Instructor: Modality: Open Book Exam
Date: Saturday 13 November, 2021   Time: 06:00pm-8:30pm
Two and half hours (2:30), Marks: 25

Question: 1	CO1: Analyze and calculate **time complexity** and **space complexity** of various algorithms or any written code using mathematical formula and comparison of algorithms.	[Marks: 5]

a)   Suppose your ID is 193-15-12479, then **X will be 7 * 9 = 63**.   [3]

How many times "Programming is fun" will be printed in the output
screen? What is the **time complexity** of the following code segment?

```
#include<stdio.h>
int main(){
    int i, j, k;

    for(i = 1 ; i <= X; i++){
        for(j = 1 ; j <= X ; j++){
            printf("Programming is fun");
        }
    }

    for(k = 1; k <= 100 ; k++){
        printf("BLC");
    }
}
```

b)   Suppose you are the system architect of a software development team and   [2]
you need a particular problem solved. Faisal and Shuvo are two software
engineers working in your team and each of them has written a correct
solution for the problem. Faisal's solution runs in $O(M^3 * N)$ while Shuvo's
solution runs in $O(N^2 * M)$ for input sizes of M & N. Which of the two
solutions do you think will run faster? Explain in no more than 3 sentences.

Question: 2	CO2: Generate and interpret the **output** of **iterative** and **recursive** codes with the analysis of the problem definition.	[Marks: 5]

a)   Write down the output of the following code segment.   **[2]**

Here, ID is the last digit of your student id + 5.

If your student id is 111-15-1059 then here ID is equal to 9+5 = 15

```c
#include <stdio.h>
int main()
{
    int i, ID;
    for(i = ID; i>=0 ; i--)
    {
      if(i%3==0)
      {
          printf("\n Start");
      }
      else
      {
          printf("\n End");
      }
      printf("\n Index is %d",i);
    }
    return 0;
}
```

b)   Write an algorithm that will find all the **Common divisors** and also the **greatest common divisor** of two integers A and B where A is the **last two** digits of your ID and B is the **reverse order** of last two digits of your ID. For example, if your ID is 123-15-1040 then A = 40 and B= 04.   **[3]**

[N.B: C.D of 40 and 4 = 1, 4 and G.C.D of 40 and 4 = 4]

What will be the **time complexity** of your code?

Question: 3	CO3: Identify which algorithm listed under which algorithmic paradigm. Compare among various algorithms/implemented codes and choose the efficient one.	[Marks: 5]

a) Given **n** groups of people of different sizes **G1, G2, …. Gn** and minibuses with capacity **C**, your task is to apply a greedy algorithm to assign each group to a minibus such that number of total required minibuses is minimized. You can assume that all group sizes are smaller than or equal to the minibus capacity. You also have to assign each groups in such a way that all members of same group stays together. **[3]**

Now, write your student Id and consider each digit of your Id as a group size and the capacity of each minibus is **9**. Find out how many minibuses you need to accommodate all the groups.

[N.B: Your id is **201-15-14564** then you have **10** groups of size **2, 0, 1, 1, 5, 1, 4, 5, 6** and **4**. The capacity of each minibus is **9**]

b) You are trying to sort an array of 1000 integers in a program running on a micro-controller where there is very little memory available but you need to complete the sorting in fastest possible time. Which sorting algorithm should you use? Explain in no more than 5 sentences. **[2]**

Question: 4	CO4: Break down and describe the **simulation** of various algorithms for different input values.	[Marks: 10]

a) Write down your mother's name and father's name. Now you need to find the **variable length code** of the characters of these two names using **Huffman coding** algorithm. You also need to find the **fixed length code** for the characters. Then calculate the size of both coding system of the names in number of bits. **[6]**

b) Sorting is an integral part of Algorithm course. There are quite a few sorting algorithms such as bubble sort, insertion sort, quick sort, merge sort etc. They are often quite interesting because of their different working methodologies and complexities. Now you have to sort the digits of your own DIU student ID using **Quick Sort** algorithm. Show each necessary steps? **[4]**

[N.B: Your id is 201-15-14564 then your array will be {2,0,1,1,5,1,4,5,6,4,} which you need to sort in the form {0,1,1,1,2,4,4,5,5,6}.]