Bahria University, Islamabad

(Department of Computer Science)

Mid Term Examination

Class/Section: BSCS-5 A and B, MSCS-0 S

(Fall 2020 Semester)

Subject: Design and Analysis of Algorithms	**Date**: 14-Dec-2020
Course Code: CSC-321	**Time**: Session II
Instructor: Saima Jawad	**Max. Marks**: 25
Time Allowed: 90 Mins	**Total Pages**: 2

Instructions:

- Submit a **single PDF file** with complete solution of all the questions.
- The exam is an individual effort and is assumed to be completed with academic honesty.
- **Plagiarism** (copying) is not tolerable and will be considered equivalent to cheating in a regular mid-term exam.
- Submission must be made on **LMS before the specified time.** Submission cannot be accepted through any other medium.
- Submitted solutions will be scored based on meeting the task requirements.
- Write your full name and enrolment number on your submission.

Name: _____ Enrolment No: _____

(USE CAPITAL LETTERS)

Q. 1 Answer the following with complete working to support your answer. **(15, 3 each)**

a) Arrange the following functions in the **ascending** order of growth rate:
$5n!, \log_2 n, 2n^3+4, 7n^2, 2^n, 10n$

b) Find the tight bound **(Θ)** of $f(n) = 3n^2\log_2 n$.

c) Suppose that a computer takes **10 micro seconds** to perform one comparison. How much time **Merge** sort will take to sort **5 million numbers**?

d) What does following algorithm compute? Formulate a recurrence relation for the algorithm's key operation count and solve it.

```
ALGORITHM Guess(A, n)
//Input: An array A[0..n – 1] of n numbers
if n = 1
      return A[0]
else re ← Guess(A, n-1)
if res ≤ A[n – 1]
      return res
else
      return A[n – 1]
```

e) What does following algorithm compute? Find the frequency of the key operation as a function of n.

```
ALGORITHM Secret(M, n)
//Input: A matrix M[1..n,1..n] of numbers of size n x n
for i ← 1 to n – 1 do
      for j  ← i + 1 to n do
            if M[i, j ] ≠ M[j, i] return false
return true
```

Q. 2 **(5+5)**

a) Sort the following numbers in **descending** order using **Quick** sort. How many **comparisons** are performed in total? Show complete working.

9, 42, 81, 24, 50, 28, 14

b) Compute the time complexity of following algorithm and trace the algorithm's working for A = {x, y, z}.

```
ALGORITHM Enigma(A, n)
//Input: An array A[1..n] of n elements
if n = 1
      write A
else
      for i← 1 to n do
            Enigma(n – 1)
            if n is odd
                  swap (A[1], A[n])
            else swap (A[i], A[n])
```

End of Question Paper