COMSATS UNIVERSITY ISLAMABAD
DEPARTMENT OF COMPUTER SCIENCE
Final Term Examination, Fall 2024

Class: BCS-V(A/B), BCT-4/5, BDS-5, BAI-5 Date: ----------
Subject: Design and Analysis of Algorithm
Instructor: Dr. Maimoona Malik/Mr. Tanveer Ahmed//Mr. Osama Subhani/ Ms. Hilal Jan
Total Time Allowed: 180 minutes Max Marks: 50
Name: _____ Registration Number: _____

NOTE: Do as directed don't go into extra detail. Only crisp and concise answers will earn you credit

- *Students are required to display their ID cards.*
- *Attempt all questions.* ***Return this question paper with your answer sheet.***
- *Use of Mobile Phones and Digital Diaries not allowed*

(CLO-1: Demonstrate fundamental concepts of algorithm along with an algorithmic approach to a given problem.)

Q # 01 **Marks = [5+5= 10]**

1.1. Identify which properties of an algorithm – ***input, output, precision, determinism, finiteness, correctness,*** and ***generality*** – does following procedure have and which is lacking? Justify your answer.

PROCEDURE: Goldbach_Conjecture ( )
// The purpose of this procedure is to check whether every even number greater than 2 is the sum of two prime numbers or not.

```
1. n ← 4
2. if( n ≠ (p1+p2)) then   // p1 and p2 are two prime numbers
3.    print " NO"
4.    break
5. else
6.    n ← n+ 2
7.    goto 2
8. print "YES"
9. end
```

1.2. Trace (Dry run) the following algorithm with m = 24 and n =60. Explain what does this algorithm compute?

ALGORITHM *Enigma*(*m*, *n*)
// Input: Two non-negative integers *m* and *n*, not both zero.
// Output: Greatest Common Divisor of *m* and *n*

```
1       if m = 0 then
2       ans ← n
3       else if m > n then
4       ans ← Enigma (n, m)
5       else
6       nLess ← n- m
7       ans ← Enigma (m, nLess)
8       return ans
```

(CLO-2: Design new algorithms for different computational problems.)

Q # 02 **Marks = [5+5+5]**

2.1. In a warehouse management system, the inventory data is stored in a 2D array M of size $m_1 \times m_2$. Each row and column in the array is sorted in ascending order to optimize the lookup of items. Each element in M represents the stock quantity of a specific product. Given a target stock quantity k, you are tasked with determining whether the stock quantity exists in M.

Problem Statement:

Design an algorithm using the **Decrease (by a constant factor) and Conquer** strategy to search for k in the 2D array M. If k is found, return its position (i,j), where i is the row index and j is the column index. If k is not found, return "Not Found."

Example:

$$M = \begin{bmatrix} 10 & 20 & 30 & 40 \\ 15 & 25 & 35 & 45 \\ 27 & 29 & 37 & 48 \\ 32 & 33 & 39 & 50 \end{bmatrix} \text{ and } k = 39$$

the algorithm should return (3,2)

If k = 70, the algorithm should return "Not Found."

2.2. Suppose you are working on a spell checker module of a software. Your program manager assigns you the following task:

Task: Let $W = w_1 w_2 \ldots\ldots w_n$ be the misspell word of length n and $C = c_1 c_2 \ldots\ldots c_m$ be the possible correct spell of W. Transform W into C with a minimum number of operations of the following types: delete a character from A, insert a character into A, or change some character in A.

The minimal number of such operations required to transform A into B is called the edit distance between A and B. Design an algorithm using ***Dynamic Programming*** that finds the edit distance from *W* to *C*.

2.3. Let T be the text file that we want to be sent over a network. Before sending the given text file we want to compress it. In order to compresses it as much as possible, we would like to find a binary code that encodes the file using as few bits as possible. The idea is to construct a binary tree based on the frequencies of characters in a given file. This tree is then used to generate variable-length codes for each character with shorter codes assigned to more frequently occurring characters.

Let $c_1, c_2, c_3 \ldots\ldots\ldots\ldots\ldots, c_n$ denote the characters; $f_1, f_2, f_3 \ldots\ldots\ldots f_n$ be their frequencies respectively. Design an algorithm using ***Greedy Approach*** that efficiently encodes a set of symbols with variable-length codes, with the goal of minimizing the total number of bits required to represent a message.

(CLO-3: Prove correctness of an algorithm using loop invariant and induction.)

Q # 03 **Marks = [3+4]**

3.1. Sate the ***Loop Invariant***, I(n) of the following algorithm.

ALGORITHM-2 *secret2*(y, z)
// Input: _____
// Output: _____
$x \leftarrow 0$
while $z > 0$ **do**
    $x \leftarrow x + y * (z \% 2)$
    $y \leftarrow 2 * y$
    $z \leftarrow \lfloor z/2 \rfloor$
end while
return x

3.2. Establish the ***correctness*** of the above algorithm using loop invariant that you define in part (3.1). Show all steps i.e., ***initialization***, ***maintenance*** and ***termination***.

(CLO-4: Analyze best, average, and worst-case behaviors of an algorithm.

Q # 04 **Marks = [3+3+4]**

4.1. Find the ***time complexity*** of the following code. Use most suitable asymptotic notation among O, **Ω**, and Θ to specify the time efficiency class of the given code.

```
void fun(int n)
{
      int j = 1, i = 0;
      while (i < n)
      {
         // Some O(1) task
          i = i + j;
          j++;
      }
}
```

4.2. Analyze the worst-case behavior of the following algorithm **COUNT3** by computing its **asymptotic upper bound**.

Algorithm COUNT3
Input: $n = 2^k$, for some positive integer k.
Output: *count* = number of times Step 5 is executed.

1. $count \leftarrow 0$
2. $i \leftarrow 1$
3. **while** $i \leq n$
4. **for** $j \leftarrow 1$ **to** i
5. $count \leftarrow count + 1$
6. **end for**
7. $i \leftarrow 2i$
8. **end while**
9. **return** *count*

4.3. Analyze the worst-case behavior of the following recursive algorithm **Test**. Use appropriate asymptotic notation among O, **Ω**, and Θ to specify the time efficiency class of the given algorithm.

```
Test(n)
{
   m ← √n
   if(n = 1)
     return
   else
   {
      while(n < 1))
      {
        Print(n)
        n ← n/2
      }
    result←Test(m)+ Test(m)
   }
  return result
}
```

(CLO-5: Explain the concept of various complexity classes with examples.)

Q # 05 **Marks = [3+2+3]**

5.1. Given the following two scenarios, classify each as either **tractable** or **intractable**, and justify your reasoning:
- **Scenario 1**: Finding the shortest path between two locations on a road map.
- **Scenario 2**: Determining the shortest possible route that visits each delivery location exactly once and returns to the starting point.

5.2. What does it mean for a problem to be NP-Complete? How is the concept of NP-Completeness related to the notion of polynomial-time reductions?

5.3. Let P_1 be an **NP-Complete** problem and P_2 and P_3 be two other problems not known to be in NP. P_2 is polynomial time reducible to P_1 and P_1 is polynomial-time reducible to P_3. Explain which complexity class problem P_3 belongs to? Justify your answer.

Best of Luck