



COMSATS UNIVERSITY ISLAMABAD
DEPARTMENT OF COMPUTER SCIENCE
Mid Term Examination, Spring 2024

Class/Section: -BSCS – 5A/B, BDS-5, BCT-6, BAI-5
Subject: -Design and Analysis of Algorithm
Instructor: -Tanveer Ahmed /Dr. Iftikhar Azim Niaz

Marks:- 25
Time:- 90 minutes
Dated:- April 17, 2024

- 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-2: Design new algorithms for different computational problems.

Question # 01: ----- **CLO-2** ----- **Marks = [7]**
Let $A[0 \text{-----} n-1]$ be an array of binary digits arranged randomly. Design an algorithm using "Brute Force" that arranges the bits in A in such an order that all 1's precede all 0's. The only constraint in arranging is that you are allowed to interchange the positions of bits if they are not similar.

Question # 02: ----- **CLO-2** ----- **Marks = [8]**
Design an algorithm by employing "Decrease (by a constant factor) and Conquer" for the following problem.
Problem Statement: Let A be a sorted array of n elements and k be the key that we want to search in A . Design an algorithm that search k in A by dividing it into three sub arrays. The size of each subarray is about $n/3$ elements. Express your solution in pseudocode.

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

Question # 03: ----- **CLO-4** ----- **Marks = [4+6=10]**
a) Find the **time complexity** of the following algorithm. Use most suitable asymptotic notation among O , Ω , and Θ to specify the time efficiency class of the given algorithm.

Algorithm Enigma(n)
{
1. $k \leftarrow n, j \leftarrow 1, i \leftarrow 0$
2. while ($k > 0$) $\checkmark O(\log_3 n)$
3. // Some $O(1)$ task
4. while ($i < n$) $n(\log_3 n)$
 a. // Some $O(1)$ task
 b. $i \leftarrow i + j$ $i \geq 1 + 2 = 3$
 c. $j \leftarrow j + 1$ $2 + 1 = 3$ $j = 2$
5. end while
6. $k \leftarrow k/3 \rightarrow \log_3 n$
7. end while
8. end

$3 > 0$
 $i = 0$
 $3/3 = 1 > 0$ $1 \neq 1$

$n = 9$
 $k = 9 > 0$
 $i = 1$ $n = 3$ $i = 3 \neq 3$
 $9/3 = 3$
 $n = 9$ $9 > 0$ $0 < n$
 $n = 18$ $18 > 0$ $i < n$ $i = 0 + 1$ $i = 1$
 $j = 2$
 $18/3 = 6$ $6 > 0$
 $6/3 = 2$ $2 > 0$
 $2/3 = 0$ $0 < n$
 $i = 1 + 2 = 3$

$6/3 = 2$ $2 > 0$ 2 times
 $3/3 = 1$

b) Consider the following Algorithm that sorts an array A recursively.

Algorithm Sort (A, p, r)

```

1  if  $p < r$  then
2     $q \leftarrow \lfloor (p+r)/3 \rfloor$ 
3    Sort (A, p, q)
4    Sort (A, q+1, r)
5    Merge (A, p, q, r)

```

Algorithm Merge(A, p, q, r)

```

1   $n_1 \leftarrow q - p + 1$ 
2   $n_2 \leftarrow r - q$ 
3  for  $i \leftarrow 1$  to  $n_1$ 
4     $L[i] \leftarrow A[p + i - 1]$ 
5  for  $j \leftarrow 1$  to  $n_2$ 
6     $R[j] \leftarrow A[q + j]$ 
7   $L[n_1 + 1] \leftarrow \infty$ 
8   $R[n_2 + 1] \leftarrow \infty$ 
9   $i \leftarrow 1$ 
10  $j \leftarrow 1$ 
11 for  $k \leftarrow p$  to  $r$ 
12   if  $L[i] \leq R[j]$  then
13      $A[k] \leftarrow L[i]$ 
14      $i \leftarrow i + 1$ 
15   else  $A[k] \leftarrow R[j]$ 
16      $j \leftarrow j + 1$ 

```

- Set up a recurrence relation for the given algorithm. **Marks = 2**
- Solve recurrence relation that you obtain in part (i) using Recursion Tree Method. Use appropriate asymptotic notation among O , Ω , and Θ to specify the time efficiency class of the given Algorithm. **Marks = 4**

Best of Luck