National University of Computer & Emerging Sciences, Karachi

Fall-2021 Department of Computer Science

Mid Term-1

11th October 2021, 01:00 PM – 02:00 PM

Course Code: CS2009	**Course Name:** Design and Analysis of Algorithm
Instructor Name / Names: Dr. Muhammad Atif Tahir, Dr. Fahad Sherwani, Dr. Farrukh Saleem, Waheed Ahmed, Waqas Sheikh, Sohail Afzal	
Student Roll No:	**Section:**

Instructions:

- Return the question paper
- Read each question completely before answering it. There are **5 questions** on **2 pages**
- In case of any ambiguity, you may make assumption. But your assumption should not contradict any statement in the question paper

Time: 60 minutes. **Max Marks: 12.5**

Question # 1 **[0.5*3 = 1.5 marks]**

Solve the following recurrences using **Master's Method.** Give argument, if the recurrence cannot be solved using Master's Method. [See appendix for Master's method 4th case if required]

$$a)\ T(n) = 6\,T\left(\frac{n}{\sqrt{n}}\right) + n + 30$$

$$b)\ T(n) = 9\,T\left(\frac{n}{3}\right) + 3n^2 + 2^3\,n$$

$$c)\ T(n) = 7T\left(\frac{n}{4}\right) + n^{\log_4 7}\log n$$

Question # 2 **[1 + 2 = 3 marks]**

Compute the time complexity of the following recurrence relations by using **Iterative Substitution Method or Recurrence-Tree Method**. [See appendix for formulas if required]

$$a)\ T(n) = 3T\left(\frac{2n}{3}\right) + n\ , \qquad \text{Assume } T(1) = 1$$

$$b)\ T(n) = T\left(\frac{n}{4}\right) + T\left(\frac{n}{2}\right) + n^2, \qquad \text{Assume } T(1) = 1$$

Question # 3 **[2 marks]**

Consider the given recurrence relation. You need to apply **Substitution Guess and Test method** on both guess one by one to find correct one.

$$T(n) = 8T\left(\frac{n}{2}\right) + n^2$$

$$\text{Guess 1}:\ T(n) = O(n^2 \log n)\ ,\quad \text{Guess 2}:\ T(n) = O(n^3 \log n)$$

<u>Question # 4</u> **[0.5+1.5=2 marks]**

Consider below given bubble sort algorithm :

```
BUBBLESORT(A)
1  for i = 1 to A.length − 1
2      for j = A.length downto i + 1
3          if A[j] < A[j − 1]
4              exchange A[j] with A[j − 1]
```

a) Let *A*’ denote the output of BUBBLESORT(A). To prove that BUBBLESORT is correct, we need to prove that it terminates and that:

$$A'[1] \leq A'[2] \leq \cdots \leq A'[n]$$

where $n = A.length$. In order to show that BUBBLESORT actually sorts, what else do we need to prove?

b) Prove below given loop invariant property for inner loop (lines 2 to 4)
<u>Loop Invariant Property of inner loop</u>: At the start of each iteration, the position of the smallest element of A[i…….n] is at most *j*

<u>Question # 5</u> **[4 marks]**

Given a sorted array, integer ***k*** and target ***t*** as input, the objective is to find ***k*** closest elements to ***t*** in the array

For example:

Input array = [17,18,20,25,30], ***k*** = 2 , ***t*** =16

Output = [17,18]

If the target is smaller than all the elements in the array then return first ***k*** elements, likewise, if target is greater than all the elements in the array then return the last ***k*** elements. The ordering of returned numbers should be maintained as in original array. Design algorithm for the above scenario that takes no longer than ***O(k + log n)*** time.

<u>Appendix</u>

Masters Theorem 4th Case

$$\text{If } f(n) \in \Theta(n^{\log_b a} \log^k n) \text{ for some } k \geq 0 \text{ then}$$

$$T(n) \in \Theta(n^{\log_b a} \log^{k+1} n)$$

$\sum_{k=0}^{\infty} ar^k = \frac{a}{1-r}$ (if r<1)

$\sum_{k=0}^{n} 2^k = 2^{k+1} - 1$