National University of Computer and Emerging Sciences, Lahore Campus

Course:	Design & Analysis of Algorithms	Course Code:	CS-2009
Program:	BS (Computer/Data Science)	Semester:	Fall 2023
Duration:	60 Minutes	Total Marks:	19
Paper Date:	2-Oct-22	Section:	ALL
Exam:	Midterm 1	Page(s):	5
Name		Roll Number	

Instruction/Notes: Solve it on question paper

Question	1	2-4	5	Total
Marks	/4	/7	/8	/19

Q1) Multiple Choice Questions [4 Marks]

a) Which of the following sort algorithms are stable?

i. Quick Sort
ii. Merge Sort
iii. Insertion Sort
iv. Count Sort

b) Which of the following sort algorithms are guaranteed to be O(n log n) even in the worst case?

i. Quick Sort
ii. Merge Sort
iii. Insertion Sort

c) Suppose we are sorting an array of eight integers using Quick sort, and we have just finished the first partitioning with the array looking like this:

2 5 0 6 8 13 9 10

Which statement is correct?

i. The pivot could be either the 6 or the 8.
ii. The pivot could be the 6, but it is not the 8.
iii. The pivot is not the 6, but it could be the 8.
iv. Neither the 6 nor the 8 is the pivot.

d) $f(n) = 30*2^n + 15*4^n + 3*16^n$

which of the following statements are true about f(N)

i. $f(n) = O(4^{2n})$
ii. $f(n) = O(8^{n+2})$
iii. $f(n) = O(2^{4n})$
iv. $f(n) = O(4^{n+4})$

Q2) What is asymptotic time complexity of following code. n is size of input. [2 Marks]

```
for(i=1; i<=n; i++)
{
   for(j=1; j<=n; j=j*2)
   {
      Arr[i] = Arr[i] + (Arr[j]*0.1)
    }
  }
```

Solution

O(nlgn)

Q3) Consider the following algorithm

```
splitData(Arr[], left, right)
{
   if(left < right)
   {
      splitData(Arr, left, (left+right)/2)
      splitData(Arr, (left+right)/2 + 1, right)
      for(i=left; i<=right; i++)
      {
         for(j=1; j<=i; j++)
         {
            Arr[i] = Arr[i] + (Arr[j]*0.1)
         }
      }
   }
}
```

Give the recurrence for the worst-case running time of above algorithm. Only write the recurrence, you do not need to solve it [2 Marks]

Solution

$T(n) = 2T(n/2) + O(n^2)$

Q4) Solve the following recurrence and write time complexity in asymptotic notation. Show all working [3 Marks]

$T(n) = 3T(n/3) + n^2$

Solution

Geometric series of n^2 { $(1/3)^0$+ $(1/3)^1$ + $(1/3)^2$ + $(1/3)^3$ + …. $(1/3)^{lg3n}$ } // Assuming series goes till infinity and x = 1/3 which is less than 1

$\leq n^2 (1 / (1 - 1/3))$

$=O(n^2)$

Q5) [8 Marks] There are 2 sorted arrays **A** and **B** of size n each. Write an algorithm to find the median of the array obtained after merging the above 2 arrays (i.e. array of length 2n).

```
Input : ar1[] = {1, 12, 15, 26, 38}
        ar2[] = {2, 13, 17, 30, 45}
Output : 16

Explanation :
After merging two arrays, we get
{1, 2, 12, 13, 15. 17, 26, 30, 38, 45}
Middle two elements are 15 and 17
Average of middle elements is (15 + 17)/2
which is equal to 16
```

(a) [4 Marks] Write an algorithm to solve the above problem that takes O(n) time.

Solution

Apply merge routine of merge sort and then find the middle element from the sorted array.

https://www.enjoyalgorithms.com/blog/median-of-two-sorted-arrays

(b) [4 Marks] Write an algorithm to solve the above problem that takes O(lg n).
Hint: You do not need to merge the two arrays. Use divide and conquer approach.

Finding median of two arrays and taking its average does not give median for sorted merged array. Here is an example

Array 1 = {1, 3, 4, 4, 4, 5, 6, 6, 6}

Array 2 = {38, 45, 66, 68, 88, 90, 91, 92, 93}

Median of first array is 4 and second array is 88, but (4+88)/2 = 46 is not the median.

Median is (38+6)/2 = 22

You do not need to apply binary search to find median or middle value of array so all solutions that used binary search to find median or middle value are incorrect as the arrays are already sorted so middle value will be median of the array.

The right solution was to get the median of the two arrays and then compare them

The solution is given in the following link

https://www.enjoyalgorithms.com/blog/median-of-two-sorted-arrays