National University of Computer and Emerging Sciences

Design and Analysis of Algorithms (CS2009)

Date: Feb 29, 2024

Course Instructor(s)

Dr. Saira Karim
Dr. Aasim Qureshi
Ms. Abeeda Akram
Mr. Usama Hassan Alvi
Mr. Sajid Ali Kazim
Mr. Iteza Muzaffar

Sessional-I Exam

Total Time: 1 Hours
Total Marks: 20
Total Questions: 3

Semester: SP-2024
Campus: Lahore
Dept: Computer Science

_____	_____	_____	_____
Student Name	Roll No	Section	Student Signature

_____	_____
Vetted by	Vetter Signature

Instruction/Notes: Attempt all questions. Answer in the space provided. **Do not attach rough sheets with this exam**. Do not use pencil or red ink to answer the questions**.** In case of confusion or ambiguity make a reasonable assumption.

CLO #1: Design algorithms using different algorithms design techniques i.e. Brute Force, Divide and Conquer, Dynamic Programming, Greedy Algorithms and apply them to solve problems in the domain of the program

Question1: **[10 marks]**

You are given a sorted array A of n integers, a search key v, and a positive integer k < n. Your goal is to find the k nearest values to the search key present in the array. Note that the search key may or may not be present in the array. Devise an efficient algorithm that returns the subarray (starting and ending index) containing the k nearest values. Also compute the time complexity of your algorithm. Please note that less effeicient solutions will be awarded lesser marks.

Input: A = { 2, 3, 5, 6, 9, 15, 23}, v=6 and k=4
Output: Subarray {3, 5, 6, 9}
Input: A= {6, 8, 9, 12, 23}, v = 4 and k=3
Output: Subarray {6, 8, 9}

CLO 2: Analyze the time and space complexity of different algorithms by using standard asymptotic notations for recursive and non-recursive algorithms.

Question2: **[5 marks]**

Derive the recurrence of the given recursive algorithm and solve the recurrence using tree method.

```
//Assume that "left" and "right" were initially pointing to the 1st and last index of the array.
data_partition(A[], left, right, N)
{
  if(left < right) {
        k = 2*((right-left+1)/7)
        v1 = data_partition(A, left, k, k)
        v2 = data_partition(A, k+1, right, N-k)
        v3 = update(A, left, right)
        return v1+v2+v3
  }
  return A[right]
}
update(A[], left, right){
        i = left
        r = 0
        for(i=left; i<=right; i*=2){
                for(m=left; m<=i; m++){
                        A[m] = A[i]*A[m]
                        r = A[m]
                }
        }
        return r
}
```

CLO 4: Implement the algorithms, compare the implementations empirically, and apply fundamental algorithms knowledge to solve practical problems related to the program.

Question3: **[5 marks]**

Suppose that you somehow know that the number of inversions in an array of size N is 10 where 10 much smaller than N. Which of the following sorting algorithms should you use to sort this array completely: Merge Sort, Selection Sort, Insertion Sort. Give an appropriate reason for your choice.