Registration Number :

VIT

Vellore Institute of Technology

CHENNAI

Continuous Assessment Test I - August 2024

Programme	B.Tech.(CSE)	Semester	Fall 2024-25
Course	Design and Analysis of Algorithms	Code	BCSE 204L
Faculty	Dr.L.Jeganathan, Dr M Janaki Meena, Dr M Raja, Dr R Sivakami, Dr B Indira , Dr G Kavipriya, Dr Jeipratha P N	Slot/Class No.	A1/CH2024250101354 /CH2024250101360 /CH2024250102306 /CH2024250100952 /CH2024250100957 /CH2024250100961 /CH2024250100543
Time	90 Minutes	Max. Marks	50

Instructions:

- Answer all the FIVE questions.
- If any assumptions are required, assume the same and mention those assumptions in the answer script.
- Use of intelligence is highly appreciated.
- Your answer for all the questions should have both the 'design' component and the 'analysis component'
- The 'Design' component should consist: understanding of the problem, logic to develop the pseudocode, illustration, pseudocode.
- The 'Analysis' component should consist: Proof-of-Correctness, Computation of T(n), Time-complexity.

---

1. **Given an array A of size n with elements from the set { 3 , -3}, Design a pseudocode to compute the length of the longest contiguous subarray whose sum is greater than 6.** For example, If A = [-3, 3, 3, 3,-3,], the length of the longest contiguous subarray whose sum is greater than 6, is 3. [10 marks]

   [**Rubrics**: Logic for pseudocode: 2 marks, Illustration for pseudocode : 2 marks, Pseudocode : 3 marks, Proof-of-Correctness: 2 marks , Time-complexity :1 mark]

2. Alphanumeric words are the words that consist of charcaters from [A-Z] or [a-z] or [0-9]. **Given an alphanumeric word W, deisgn a pseudocode to arrange the characters of W in to a new alphanumeric word W' such that all the numeric characters of W' occur in an increasing order, all the alphabetic characters of W' occur in a decreasing order and all the positions of W which have the alphabetic characters, have the alphabetic characters in W' also, all the positions of W which have the numerical characters, have the numerical characters in W' also.** For example, if $W = ab17X6$, then your pseudocode should output $W' = Xb16a7$. [10 marks]

   [**Rubrics**: Logic for pseudocode: 2 marks, Illustration for pseudocode : 3 marks, Pseudocode : 3 marks, Time-complexity :2 mark]

3. Given an array A of integers, we assign a value called as Maximum-sum-Sub-Array value (denoted as $MSA(A)$) which is the maximum value among the sum of all the contiguous subarrays (i.e., a subarray with consecutive elements) in A. For the array $A = [1, -2, 3, 4, -1, 2, 1, -5, 4]$, the subbaray $[3, 4, -1, 2, 1]$ has the maximum sum 9 and $MSA(A) = 9$.

   Consider an n-digit positive integer N. We define $MSA(N)$ as the Maximum-sum -Sub-Array value of the array of size n which has all the digits of N, in the same order of occurrence as in N. **Given n positive integers, $a_1, a_2, ..., a_n$, all a_i's not equal to zero, design a pseudocode which will output the positive integers $a'_1, a'_2, ..., a'_n$ such that $MSA(a'_1) \geq MSA(a'_2) \geq ... \geq MSA(a'_n)$, where the relation '$\geq$' is the usual'greater than or equal to' relation and $a'_i \in \{a_1, a_2, a_3.., a_n\}$, for all i.**

That is, your pseudocode should arrange the given numbers in a decreasing order of their MSA-values. [10 marks]

[**Rubrics**: Logic for pseudocode: 2 marks, Illustration for pseudocode : 2 marks, Pseudocode : 3 marks, Proof-of-Correctness: 2 marks , Time-complexity :1 mark]

4.

Algorithm 1 PQRS

```
1: Input: A positive integer A
2: Output: result
3: Initialize an empty array F
4: Y ← A
5: while Y > 0 do
6:     T ← Y mod 10
7:     Add T to F
8:     Y ← Y ÷ 10
9: end while
10: n ← length of F
11: for i = 0 to n − 1 do
12:     for j = 0 to n − i − 2 do
13:         if F[j] < F[j + 1] then
14:             Y ← F[j]
15:             F[j] ← F[j + 1]
16:             F[j + 1] ← Y
17:         end if
18:     end for
19: end for
20: result ← 0
21: Z ← 1
22: for i = n − 1 to 0 by step −1 do
23:     result ← result + F[i] × Z
24:     Z ← Z × 10
25: end for
26: Return result
```

Understand the functionality of the above algorithm and answer the following.

(a) Identify an input, which when fed to the above algorithm, returns that input itself as the output. [2 marks]

(b) Describe the functionality of the above algorithm. [3 marks]

(c) Compute the time-complexity of the algorithm. [2 marks]

(d) Modify the above algorithm in such a way that the time-complexity of the modified algorithm is better than the above algorithm. [3 marks]

5. A problem called 'Rod Assembly Problem' (RAP) is described as follows: Given n rods with a description $(L_1, p_i 1), (L_2, p_2), ..., (L_n, p_n)$, where L_i represents the length of the rod i units and p_i represents the price of the rod of length i units. We can assemble these rods and make a rod of bigger length. Given the description of all the rods and the target length T units, task is to identify the rods that can be assembled , to make a bigger rod of length T units in such way that the cost of assembling the rod of Length T is minimal. Note that the cost involved in the assembling process is the prices of the rods that are involved in the assembly. For example, if the inputs are $\{(2,3),(3,5),(5,7),(8,10)\}$ and the target length 10, solution is $\{(2,3),(8,10)\}$ which means that, the optimal way to assemble a rod of length 10 units is to combine the rods length 2 and 8 units. **Given the required inputs, design a dynamic programing based pseudocode for the 'Rod Assembly Problem'.** [10 marks]

[**Rubrics**: Logic for pseudocode: 2 marks, Illustration for pseudocode : 3 marks, Pseudocode : 3 marks , Time-complexity :2 marks]

Registration Number: 22BRS1017

VIT

Vellore Institute of Technology
CHENNAI

Continuous Assessment Test I - August 2024

Programme	B.Tech.(CSE)	Semester	Fall 2024-25
Course	Design and Analysis of Algorithms	Code	BCSE 204L
Faculty	Dr B Indira, Dr J Omana, Dr M Raja, Dr N Sivarakrishnan, Dr D Selvam, Dr G Kavipriya	Slot/Class No.	A2/CH2024250100959 /CH2024250101375 /CH2024250101371 /CH2024250102307 /CH2024250100963 /CH2024250101368
Time	90 Minutes	Max. Marks	50

Instructions:

- Answer all the FIVE questions.
- If any assumptions are required, assume the same and mention those assumptions in the answer script.
- Use of intelligence is highly appreciated.
- Your answer for all the questions should have both the 'design' component and the 'analysis component'
- The 'Design' component should consist: understanding of the problem, logic to develop the pseudocode, illustration, pseudocode.
- The 'Analysis' component should consist: Proof-of-Correctness, Computation of T(n), Time-complexity.

---

1. Consider the following recursive algorithm:

Algorithm 1 *RecSum*(n)

```
Input: A positive integer n > 0
Output: Print n
if n == 1 then
    return 1
else
    return (n + RecSum(n − 1))
end if
```

Understand the functionality of the above algorithm and answer the following.

(a) Trace the algorithm by assuming n = 5 and illustrates the description of the algorithm that you computes. [3 Marks]

(b) Determine the running time of the recursive algorithm RecSum(n). [2 Marks]

(c) Modify the algorithm to follow an iterative procedure to solve the same problem. [3 Marks]

(d) Compute the running time of your modified iterative algorithm and compare it with the recursive version. [2 Marks]

2. Consider two files X and Y of different sizes. Each file is represented by a sequence of characters where changes are often made in adjacent pairs of lines (e.g., function names or control structures). Your task is to find the **Longest Common Pair Subsequence (LCPS)** between these two sequences to identify common blocks of code. For the file X = IFELSEFORIFRETURNN and Y = FORIFRETURNELSEIF the LCPS identifies the common blocks of code between the above two files could be FORIFRETURN. **Design a pseudocode to solve the Longest Common Pair Subsequence (LCPS) problem which follows a dynamic programming strategy.** Your pseudocode should return the longest matching subsequence for the above given example. [10 marks]

[**Rubrics**: Logic for pseudocode: 2 marks, Illustration for pseudocode : 2 marks, Pseudocode : 3 marks, Proof-of-Correctness: 2 marks , Time-complexity :1 mark]

3. Let $A_1, A_2, \cdots A_n$ be n matrices, where A_i size is $p_{i-1} \times p_i$ for $i = 1, 2 \cdots n$. Let C be a cost matrix of size $n \times n$, where $C[i][j]$ is the minimum number of scalar multiplications to compute the chain $A_i \times A_{i+1} \times \cdots \times A_j$. Design a dynamic programming-based algorithm to compute the cost matrix C. [10 marks]

[**Rubrics**: Pseudocode: 4 marks, Illustration for pseudocode : 4 marks, Time-complexity :2 mark]

4. Given an array A of integers, and your task is to find a contiguous subarray that maximizes the sum. The subarray must have at least p positive elements and at least q negative elements. For example: $A = [4, -1, 2, 1, -5, 4, -3, 2, -1, 5, -6]$, $p = 2$ (minimum positive elements) , $q = 2$ (minimum negative elements). Design an algorithm to find the contiguous subarray that maximizes the sum $\sum_{i=j}^{k} A[i]$ and the algorithm should return the starting and ending indices of the subarray that meets the criteria, along with the maximum sum. [10 marks]

[**Rubrics**: Logic for pseudocode: 2 marks, Illustration: 3 marks, Pseudocode : 4 marks, Time-complexity :1 mark]

5. Consider an array A of n elements where each element may occur one or more times. For example: A[12, 45, 67, 12, 89, 67, 45, 12, 45, 89, 67, 45, 12, 89, 67, 89, 45], where
12 has occurred 4 times
45 has occurred 5 times
67 has occurred 4 times
89 has occurred 4 times
The expected output should be: [12, 67, 89, 45].
Design an algorithm to return the count of the number of occurrences of each element in the given set. Design a divide and conquer algorithm to arrange the elements in ascending order according to their count occurrence count. If two elements have the same count, they should be arranged in ascending order based on their value. Analyze the time complexity of the algorithm. [10 marks]

[**Rubrics**: Logic for pseudocode: 2 marks, Illustration for pseudocode : 2 marks, Pseudocode : 3 marks, Proof-of-Correctness: 2 marks , Time-complexity :1 mark]

$\Longleftrightarrow \Longleftrightarrow \Longleftrightarrow \Longleftrightarrow$

Registration Number : 22BCE1351

VIT
Vellore Institute of Technology

Continuous Assessment Test I - August 2024

Programme	B.Tech.(CSE)	Semester	Fall 2024-25
Course	Design and Analysis of Algorithms	Code	BCSE 204L
Faculty	Dr.Srinivasa Rao U, Dr. Jayaram B, Dr.Tapabrata Roy, Dr. Sivapriya Dr. Lekshmi	Slot/Class No.	D2/CH2024250101207 /CH2024250101210 /CH2024250102302 /CH2024250101209 /CH2024250102301
Time	90 Minutes	Max. Marks	50

Instructions:

- Answer all the FOUR questions.
- If any assumptions are required, assume the same and mention those assumptions in the answer script.
- Use of intelligence is highly appreciated.
- Your answer for all the questions should have both the 'design' component and the 'analysis component'
- The 'Design' component should consist: understanding of the problem, logic to develop the pseudocode, illustration, pseudocode.
- The 'Analysis' component should consist: Proof-of-Correctness, Computation of T(n), Time-complexity.

---

1. Let α be a comparison operator that denotes an inequality between positive two numbers. The operator α is typically placed between two numbers being compared and signifies that the sum of digits of the first number is less than the sum of digits of the second number. If two numbers have the same sum of digits, the smallest number comes before the largest number. For example , $222\alpha199$ is true , because $6(=2+2+2) \leq 19(=1+9+9)$. But $98\alpha111$ is not true, because $17(=9+8) \not< 3(=1+1+1)$.
Alpha Sort problem: Let S be an array of n positive integers. Rearrange the given array S of elements according to the comparison operator α on the elements. For example, let $S = [22, 1111, 11, 9]$. The resultant output should be $[11, 22, 1111, 9]$.
Design two different algorithms using two different design techniques to solve the **Alpha Sort problem**. For the example given above, your algorithms should return $[11, 22, 1111, 9]$. As a result, justify which of the two design techniques is more efficient to solve this problem.

[15 marks]

[**Rubrics**: Logic for both techniques: 2 marks, Illustration for each Logic:2 marks, Algorithms: 8 marks , Time-complexities: 3 marks]

2. A subarray of an array, is an array that is also a contiguous part of the array. In any given array, the maximum sum subarray is a series of contiguous elements with the maximum sum. For instance, in the below array, the highlighted subarray(in bold) has the maximum sum (4-1+2+1=6):

-3	1	-8	4	-1	2	1	-5	5

2D Maximum Sum Subarray problem: Let A be a two-dimensional array of size $n \times n$. Let's assume that R_1 represents the first row, R_2 represents the second row ,$\cdots$, and R_n represents the n^{th} row in A.

Your task is to separately compute the maximum subarray sum for each row of the array A. Design two different algorithms using two different design techniques to solve the **2D Maximum Subarray Sum problem**. As a result, justify which of the two design techniques is more efficient to solve this problem.

[15 marks]

[**Rubrics**: Logic: 2 marks, Illustrations: 2 mark, Algorithms :8(4+4) marks and Time-complexities :3 marks]

3. Consider the following algorithm:

Algorithm 1 M-Dist

```
1: Input: Array of A[0, 1, ..., n − 1] of numbers and |x| is the absolute value of x
2: Output: dmn
3: dmn ← ∞
4: for i = 0 to n − 1 do
5:    for j = 0 to n − 1 do
6:       if i ≠ j and |A[i] − A[j])| < dmn then
7:          dmn← |A[i] − A[j])|
8:       end if
9:    end for
10: end for
11: return dmn
```

Understand the functionality of the above algorithm and answer the following: [10 marks]

(a) Write the output of **M-Dist** algorithm, when $A = [12, 1 - 4, -5, -6]$. [2]

(b) Describe the functionality of **M-Dist** algorithm and also write the proof of correctness for the algorithm. [1+4]

(c) Compute the time-complexity of the algorithm. [3]

4. Given an array $A[1 \ldots n]$ of integers. Design an algorithm with time complexity $\Theta(nlogn)$ to check if the elements of A are arranged in non-decreasing order of their values. [10 marks]

[**Rubrics**: Logic: 2 marks, Illustration : 1 mark, Algorithm : 5 marks and Time-complexity :2 marks]

$\Longleftrightarrow \Longleftrightarrow \Longleftrightarrow \Longleftrightarrow$

VIT
Vellore Institute of Technology

Continuous Assessment Test I - January 2023

Programme	B.Tech.(CSE)	Semester	Winter 2022-23
Course	Design and Analysis of Algorithms	Code	BCSE 204L
Faculty	Dr.L.Jeganathan, Dr B Jayaram, Dr M Janaki Meena, Dr Muthukumaran, Dr Rajkumar Arul, Dr R Ramesh	Slot/Class No.	B1/CH2022235000277 /CH2022235000279 /CH2022235000280 /CH2022235000283 /CH2022235000284 /CH2022235000285
Time	90 Minutes	Max. Marks	50

Instructions:

- Answer all the FIVE questions.
- If any assumptions are required, assume the same and mention those assumptions in the answer script.
- Use of intelligence is highly appreciated.
- Your answer for all the questions should have both the 'design' component and the 'analysis component'
- The 'Design' component should consist: understanding of the problem, logic to develop the pseudocode, illustration, pseudocode.
- The 'Analysis' component should consist: Proof-of-correctness, Computation of T(n), Time-complexity.
- For question numbers. 1 & 2, rubric is : Logic (2 marks), Illustration(2 marks), Pseudocode(2 marks), Proof-of-correctness(2 marks), running-time and the time-complexity (2 marks).
- For question no. 3, rubric is : logic of the three pseudocodes (3), illustrations for the three pseudocodes (2), three pseudocodes (2) time-complexities of the three pseudocodes (2) and the conclusion on the efficient pseudocode (1).
- For question No.5, rubric is : Proposal of the problem (1 marks), logic of the pseudocode A(1 mark), Illustration for A(1 mark), pseudocode A(1 mark), logic for the recursive pseudocode B (2 mark),pseudocode B(1 marks), Illustration for B(1 marks), Time-complexities of A and B and the conclusion (2 marks),

---

1. The symbols a, e, i, o, u are called as vowels and the remaining symbols of English alphabet are called consonants. A word of length 6 (a word with 6 symbols) w_1 is said to be $vowel - greater$ than another word w_1 of lenght 6 if the number of distinct vowels is greater than or equal to the number of distinct vowels in w_2. If two words w_1, w_2 have the same number of vowels, the word which comes later in an alphabetical order, is said to be $vowel - greater$ than the other. The number of distinct vowels in *abbaae* is 2. The word *aaeoub* is said to be $vowel - greater$ than the word *abedfg* since the number of distinct vowels in the first word is 4 and the number of distinct vowels in the second word is 2. Similarly, the word *deoudd* is $vowel - greater$ than *cedodu*. Here, both the words have the same number of vowels and the first word *deoudd* comes later in the alphabetical order when compared with the second word *cedodu*. Given the three words *dedodu*, *aeiouu*, *befghs*, they are arranged in an ascending order with respect to the operator $vowel - greater$, as *befghs*, *dedodu*, *aeiouu*. Consider an array of n distinct words of same length (ie., no two words are same), design a pseudocode to sort the array in an ascending order with respect to the relational operator $vowel - greater$. Your design component should contain all the required components. Analyse the pseudocode with all the required steps. [10]

2. Given an n-digit positive integer m (with digits from 0 to 9) design a pseudocode with the 'Divide-Conquer-Combine' strategy to generate all n-digit possible integers m' derived from m such that all digits of m are present in m' and the the first digit of m' is same as the last digit of m'. The pseudocode

should return all the possible words m' in an ascending order. For example, if m=12323, your pseudocode should return 21332, 23132, 23312, 31223, 32123, 32213. If $m = 1234$, it is clear that no m' is possible since there are no duplicate digits . Your design component should contain all the required steps. Analyse the algorithm with all the required steps. [10]

3. Given a matrix A

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ . & . & . & . \\ . & . & . & . \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

where $a'_{ij}s$ are the integers in the position (i, j). Design three different pseudocodes using the 'Divide-Conquer-Combine' strategy, in three different ways to compute M, where

$$M = Max\{a_{ij}, 1 \leq i \leq n, 1 \leq j \leq n\}.$$

Based on the time-complexities of the three pseudocodes, identify the efficient pseudocode. Besides the three different pseudocodes, your answer should contain the logic of the three pseudocodes, illustrations for the three pseudocodes, time-complexities of the three pseudocodes and your conclusion on the efficient pseudocode. [10]

4. Consider the following algorithm.

- Algorithm YYY(n)
- 1. Read n
- 2. Initialise sum to 0
- 3. Initialise digit to 1
- 4. Repeat
- 5. digit= Mod(n,10)
- 6. n=n/10
- 7. sum=sum*10+digit
- 8. Until n==0
- 9. Write sum
- 10. Stop.

Understand the functionality of the above algorithm and answer the following.

(a) Write the output of the algorithm when (i) n = 231, (ii) n= 5 [2]

(b) Describe the functionality of the above algorithm. [3]

(c) Compute the time-complexity of the algorithm. [2]

(d) Modify the above algorithm in such a way that the time-complexity of the modified algorithm remains the same. [3]

5. Propose a problem in detail (of your choice), which is not discussed in your class-room as well as in the lab sessions. Write a brute-force pseudocode A to solve the problem proposed by you. Compute the time-complexity of the pseudocode A. Transform your pseudocode A into an equivalent pseudocode B which uses divide-conquer-combine strategy. Compute the time-complexity of both the pseudocodes A and B and identify the efficient one. Note that you are required to propose a problem for which a divide-conquer-combine based pseudocode is possible. [10]

VIT

Vellore Institute of Technology

School of Computer Science and Engineering

Winter Semester 2023-2024

Continuous Assessment Test – I

Programme &Branch: B.Tech (BCB/BCE/BCI/BCT/BDS/BKT) **Slot : A1+TA1**

Course code: BCSE2041. – Design and Analysis of Algorithms

Faculty (s): ALL

Course Name (s): ALL

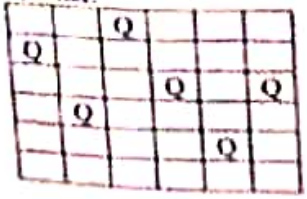
Time Duration: 90 Mins. **Max. Marks: 50**

General instruction(s): **ANSWER ALL THE QUESTIONS**

Q.No.	Question
1.	(a) Discuss the selection sort algorithm by providing its pseudo code. Discuss the loop invariant in perception with selection sort algorithm. Check the proof of correctness for the same. **(4-Marks)**
	(b) Using the master's theorem, solve the recurrence relation (i) $T(n) = 4T\left(\frac{n}{2}\right) + n^2$ (ii) $T(n) = 7T\left(\frac{n}{2}\right) + 18n^2$ **(6-Marks)**
2.	Consider the string "*sheselisseashellsbytheseashores*". Use minimum number of bits for transmitting the said string. Calculate the number of bits used to encode this using Huffman coding technique. Identify the bits required in both fixed-size and variable length encoding. **(10-Marks)**
3.	Design and develop an algorithm to multiply 2 integers and analyze their time complexity. Illustrate the technique to multiply the numbers 1334, and 1253. **(10-Marks)**
4.	Provide the optimal parenthesization while multiplying the matrices A1, A2, A3, A4, A5 having dimensions mentioned below. **(10-Marks)**

A1	2 × 5
A2	5 × 10
A3	10 × 5
A4	5 × 6
A5	6 × 10

5.	Define 6-Queens problem. Assuming that the queens are placed column-wise in the 6-Queens problem solved using backtracking. Consider the following intermediate state where the queens are attacking each other. Show the steps that involve backtracking to attain the solution where no queen attacks the other. **(10-Marks)**



VIT

Vellore Institute of Technology

School of Computer Science and Engineering

Winter Semester 2023-2024

Continuous Assessment Test – I

Programme Name &Branch: B.Tech (BCB/BCE/BCI/BCT/BDS/BKT) **Slot : A2+TA2**

Course Name & code: BCSE204L – Design and Analysis of Algorithms

Class Number (s): ALL

Faculty Name (s): ALL

Exam Duration: 90 Min. **Max. Marks: 50**

General instruction(s): ANSWER ALL THE QUESTIONS

Q.No.	Question
1.	a) Demonstrate the iteration method to compute the asymptotic complexity for the following recurrence. $T(n) = 4T\left(\frac{n}{3}\right) + n^2$ **(5-Marks)**
	b) Use master's method to compute the asymptotic complexity for the following recurrences. In each case, identify the case of master method that it uses to compute the asymptotic complexity. **(5-Marks)** (a) $T(n) = 16T\left(\frac{n}{4}\right) + n^2$ (b) $T(n) = 3T\left(\frac{n}{3}\right) + n^{1.2}$
2.	Discuss how the greedy approach is used to solve optimization problems. Construct the frequency table of characters in "Hi! How are you?" in a non-decreasing order of frequency. Use Huffman code to find the code word for each character. **(10-Marks)**
3.	Define maximum sub-array sum problem. Find the series of contiguous elements that results in the maximum sub-array sum for the array given below. **(10-Marks)**

-2	-3	1	4	-1	3	5	-4	6	1

4.	Longest common subsequence (LCS) problem is the problem of finding the longest subsequence common to all sequences in a set of sequences. Consider the sequences "ACCGGTCGAGT" and "GTCGTTCGG". Find the length of the longest common subsequence using dynamic programming approach with the pseudocode for the same. **(10-Marks)**
5.	Given a set of non-negative integers S = {3, 34, 4, 12, 5, 2} and a sum 30, determine the subsets of S, whose sum is equal to 30 using backtracking. **(10-Marks)**

Let's Connect.....!!

VIT-C 27 (Satya Helpzz) Group 1
VIT-C 27 (Satya Helpzz) Group 2

MasterLink (Everything i do)

Wanna be among first few ones to get stuffs Like this.....???

Subscribe to my Youtube channel & Join my group now ! Don't miss out!

Wanna chill ?? Shradha Didi in Chennai...!!?? (Coverup by me)

Unveiling All-Subject PYQs: Your Ultimate VIT Exam Strategy!

Youtube | Instagram | LinkedIn | Facebook