VIT

Reg. No. : 21BPS1250

Final Assessment Test (FAT) - July/August 2023

Programme	B.Tech.	Semester	Fall Inter Semester 22-23
Course Title	DESIGN AND ANALYSIS OF ALGORITHMS	Course Code	BCSE204L
Faculty Name	Prof. Dr.Rajakumar Arul	Slot	C1+TC1
		Class Nbr	CH2022232500925
Time	3 Hours	Max. Marks	100

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

Section A (4 X 10 Marks)
Answer all questions

01. Given a set of n points $\{C_1(x_1, y_1), C_2(x_2, y_2), ..., C_n(x_n, y_n)\}$ that represents cities $C_1, C_2 ... C_n$, where the x-coordinate represents the longitude and the y-coordinate represents the latitude of the respective city, design an algorithm using the Divide-Conquer-Combine strategy *(DCC)* to arrange the cities in the decreasing order of longitude. For the purpose of this problem, assume that the latitudes and longitudes are positive integers without involving any directions. When two cities have the same longitude arrange those cities based on the decreasing order of latitude. [10]

Rubrics:

Logic(2 marks), Illustration (3 marks), Pseudocode (3 marks), Running time & Time-complexity (2 marks)

02. Understand the following algorithm and answer the questions below: [10]

Algorithm 1 : $F(A)$

```
Input : A is an array of positive integers
Here '/' performs integer division
n = A.length()
for i = 1 to n do
    A[i] = F1(A[i])
end for
return A
```

Algorithm : $F_1(n_1)$

```
s1 = 0
s1 = s1*10 + n1 mod 10
n1 = n1 / 10
s1 = s1*10 + n1 mod 10
n1 = n1 / 10
n1 = n1 * 100
n1 = n1 + s1
return n1
```

(i) What will be the output of the algorithm F when the input array A = [123, 4578, 2391, 4165]? [2 Marks]

Final Assessment Test (FAT) - May 2024

Programme	B.Tech.	Semester	WINTER SEMESTER 2023 - 24
Course Title	DESIGN AND ANALYSIS OF ALGORITHMS	Course Code	BCSE204L
Faculty Name	Prof. Jayaram B	Slot	A1+TA1
		Class Nbr	CH2023240502391
Time	3 Hours	Max. Marks	100

General Instructions:

- Write only Register Number in the Question Paper where space is provided (right-side at the top) & do not write any other details.
- 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: Computation of T(n), Time-complexity.

Section - I

Answer all questions (4 X 10 Marks = 40 Marks)

01. Let n be a positive integer. Let $\Sigma_1 = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, \ldots n^2\}$. $\Sigma_2 = \{a, b, c, \ldots, z\}$. Consider a grid of size $n \times n$, with n rows and n columns **[10]**

Given a grid of size $n \times n$, design a backtracking-based pseudocode to fill the cells of the grid with elements from $\Sigma_1 \cup \Sigma_2$ such that (i) all rows start and end with elements from Σ_2 (ii) all columns start and end with elements from Σ_1 (iii) Condition (i) and Condition (ii) is exempted for the first row, last row, first column and the last column of the grid (iv) No two rows, No two columns, No two diagonals have any repetition of elements from $\Sigma_1 \cup \Sigma_2$. Your design component should contain all the required steps and the analysis component should contain all the required components.

backtracking

n queen

[**Rubrics:** Logic(2 marks), Illustration (3 marks), Pseudocode (3 marks), Running time & Time-complexity (2 marks)]

02. Given a graph $G = (V, E, s, c, t)$, where V is the set of vertices, E is the set of edges, $s \in V$ is a vertex designated as a source vertex, c is the set of capacities of all the edges of G, t is a vertex designated as the target vertex. Let $|f|$ denote the maximum flow of the network G with s as the source vertex and t as the target vertex. **Given the graph $G = (V, E, s, c, t)$, design a pseudocode to compute the maximum flow from s to t such that the total number of** **[10]**

augmenting paths computed in the pseudocode to calculate $|f|$, should not be greater than the number of edges in the graph. Your design component should contain all the required steps. Analyze the pseudocode with all the required steps.

[**Rubrics**: Logic(2 marks), Illustration (3 marks), Pseudocode (3 marks), Running time & Time-complexity (2 marks)]

03. a) Consider the problem P: Given a positive integer n, the task is to compute $n^{\log n}$. Compute the class-complexity of the problem P (P/NP/NPC) with justification. [5 marks] **[10]**
[**Rubrics** : Identification of class-complexity (2 marks), Justification (3 marks)

b) Consider the problem, **NAVIGATION** : Consider an n- digit number $a_1a_2a_3 \ldots a_n$, with distinct digits. Assume that $a_1, a_2, \ldots, a_n$ form the nodes of an undirected graph G. There will be an edge connecting a_i and a_j in G iff $i < j$. Task is to navigate from a node in G and reach the same node in such a way that, every node is visited only once except the starting and the ending node. As an example, for the number is12347, one possible NAVIGATION path is:
$1 - 2 - 3 - 4 - 7 - 1$.
Compute the class-complexity of the problem **NAVIGATION** (P/NP/NPC) with justification. [5 marks]

04. Let $\Sigma = \{a, b, c, \ldots x, y, z\}$. Let T and T_1 be any two strings with symbols from Σ. T is said to be a cousin of T_1 if T_1 can be obtained from T by shifting all the symbols(except the first symbol) of T one position to left and the first symbol of T is shifted to the rightmost position, or viceversa. The string $abcdef$ is a cousin of $bcdefa$. **Given two strings T, T_1, design a pseudocode to decide whether T and T_1 are cousins or not. Your pseudocode should use any of the string-matching pseudocodes discussed in the class.** Your design component should contain all the required steps. Analyze the pseudocode with all the required steps. [10 marks] **[10]**

string matching

[**Rubrics**: Logic(2 marks), Illustration (3 marks), Pseudocode (3 marks), Running time & Time-complexity (2 marks)]

Section - II

Answer all questions (4 X 15 Marks = 60 Marks)

05. Consider the 2-dimensional plane where the points are represented by a pair of integers. An edge with the end-points p_i and p_j is a line represented as $l(p_i, p_j)$, connecting the points p_i and p_j. A polygon $P = \{p_1, p_2, \ldots, p_m\}$ is a set of points that form a closed figure with the edges $p_1 - p_2, p_2 - p_3, p_3 - p_4, \ldots, p_{(m-1)} - p_m, p_m - p_1$. The points $p_1, p_2, \ldots p_m$ are the vertices of the polygon. A polygon is said to be simple if none of the edges of the polygon cross itself. A convex polygon is a simple polygon where, any line segment connecting two points within the polygon will lie entirely inside the polygon or on its boundary. Convex Hull of a set of points is the smallest simple convex polygon that contains all the points in the set. Circumcircle of a convex hull of the points $p_1, p_2, \ldots, p_n$, is the circle that passes through all the vertices of the convex hull. **[15]**
Given n points $\{p_1, p_2, \ldots, p_n\}$ in the 2-dimensional plane, design a pseudocode to compute the radius of the circumcircle of the convex hull of the given n points. Your 'design' component should contain all the required steps and the 'analysis' component should contain all the required steps. [15 marks]

convex hull

[**Rubrics**: Logic(4 marks), Illustration (4 marks), Pseudocode (4 marks), Running time & Time-complexity (3 marks)]

06. You are organizing a function for which you have invited n guests. Every guest will be picked up from their house and dropped at the venue of the function. [15]

Every car will start from the venue of the function to pick up the guest and drop the guest at the venue. Every guest g_i will have a pair (s_i, d_i), where s_i is the start-time of the car from the venue to pick the guest and the d_i is the drop-time of the guest g_i at the venue. **Given the details (s_i, d_i) of the guests g_i, $i = 1, 2\ldots, n$, design an algorithm to compute the minimum number of cars to be booked for the purpose.**

For example, If

$g_1 : (8:15, 9:05), g_2 : (8:40, 9:25), g_3 : (9:10, 9:45), g_4 : (9:47, 10:50), g_5 : (9:20, 10:20)$

then minimum of two cars are required. Your `design' component should contain all the required steps and the `analysis' component should contain all the required steps. [15 marks]

[**Rubrics**: Logic(4 marks), Illustration (4 marks), Pseudocode (4 marks), Running time & Time-complexity (3 marks)]

07. Given a chain of n matrices $< A_1, A_2, \ldots, A_n >$ where each matrix A_i is of size $p_{(i-1)} \times p_i, i = 1, 2, 3, \ldots n$. Product of the chain of matrices $< A_1, A_2, \ldots, A_n >$ is the value of the expression $A_1 \times A_2 \times A_3 \ldots \times A_{n-1} \times A_n$. Given an array $[p_0, p_1, p_2, \ldots, p_n]$, **Design a divide-conquer-combine pseudocode to parenthesize the chain $< A_1, A_2, \ldots, A_n >$ such that minimum number of scalar multiplications are involved in the computation of the product of the chain.** Your `design' component should contain all the required steps and the `analysis' component should contain all the required components. [15 marks] [15]

MCM

[**Rubrics**: Logic(4 marks), Illustration (4 marks), Pseudocode (4 marks), Running time & Time-complexity (3 marks)]

08. Given an unlimited supply of coins of denominations $x_1, x_2, \ldots, x_n$ and a value v, the task is to make change for the value v. That is, for the given v, we have to find a set of coins whose total value is v. If the denominations are 5 and 10, we can make the change for 15 as 10+5 and we can not make the change for 12. **Given the denominations $x_1, x_2, \ldots, x_n$ and a value v, design an dynamic programming based algorithm to express v using denominations $x_1, x_2, \ldots, x_n$ such that minimum number of coins are required .** In case, if v can not be expressed in terms of the given denominations, your algorithm should return -1. Your `design' component should contain all the required steps and the `analysis' component should contain all the required components. [15 marks] [15]

unbounded Knapsack

[**Rubrics**: Logic(4 marks), Illustration (4 marks), Pseudocode (4 marks), Running time & Time-complexity (3 marks)]

VIT
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Final Assessment Test - Model Questions

Programme	B.Tech.(CSE)	Semester	XX
Course	Design and Analysis of Algorithms	Code	BCSE 204L
Faculty	YY	Slot/Class No.	Z/CH123456789
Time	180 Minutes	Max. Marks	100

Instructions:

- Answer all the EIGHT 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: logic to develop the pseudocode, illustration, pseudocode.
- The 'Analysis' component should consist: Computation of T(n), Time-complexity.
- Rubrics for question No. 1,2 : Logic(2 marks), Illustration (3 marks), Pseudocode (3 marks), Running time & Time-complexity (2 marks)
- Rubrics for question No.3(a), 3(b): Identification of class-complexity (2 marks), Justification (3 marks)
- Rubrics for question No. 5,6,7: Logic(3 marks), Illustration (5 marks), Pseudocode (5 marks), Running time & Time-complexity (2 marks)
- Rubrics for question No. 8 : Logic for A and B (4 marks), Illustration for A and B (4 marks), Pseudocodes : A and B (4 marks), Running time & Time-complexity of A and B (3 marks)

---

1. The set of all real numbers that lie between an integer a and an integer b (a and b are also included in the set), $a < b$, is represented as $[a, b]$ and is called as interval a,b. Given $[a_1, b_1], [a_2, b_2], ...[a_n, b_n]$, design a greedy pseudocode to identify the maximum number of intervals (among the given n intervals) such that the intersection of any two intervals (among the intervals identified by you) is always empty. Your design component should contain all the required steps. Analyse the algorithm with all the required steps. [10 marks]

2. Consider the 2-dimensional plane where the points are represented by a pair of integers. A line with the end-points p_i and p_j is represented as $l(p_i, p_j)$. A polygon $P = \{p_1, p_2, ..., p_m\}$ is a set of points that form a closed figure with the edges $p_1 - p_2, p_2 - p_3, p_3 - p_4, ..., p_{(m-1)} - p_m, p_m - p_1$. Edges of the polygon can also be called as the line connecting the points. For example, the edge $p_1 - p_2$ can be called as a line l whose end-points are p_1 and p_2. A polygon is said to be a simple polygon if none of its edges cross itself. Given a polygon $P = \{p_1, p_2, ..., p_m\}$, design a pseudocode to decide whether the polygon is a simple polygon or not. [10 marks]

3. (a) Consider the problem P: Given a positive integer N, task is to check whether the sum of the factors of N except N equals N or not. For example, factors of 6 (other than 6) are 1,2,3. So, sum of the factors of 6 (other than 6) is equal to 6. Identify the complexity-class (P/NP/NPC) of the problem Q with justification. [5 marks]

   (b) Consider the problem P_1: Given a set S of positive integers, does there exist two disjoint subsets S_1, S_2 of S such that the sum of the integers in S_1 equals the sum of the intgers of S_2. For the input $S = \{3, 4, 2, , 5\}$, the solution to the problem is 'yes' since there exist two subsets $S_1 = \{3, 4\}, S_2 = \{2, 5\}$ such that the sum of the elements of S_1 equals the sum of the elements of S_2. Identify the complexity-class (P/NP/NPC) of the problem P_1 with justification. [5 marks]

4. Understand the Algorithm and answer the following questions .

- f2(B):
  - q = B mod 10
  - If (q mod 2) == 0 then return B
  - return B
  - else return 0
- f1(A, l, h)
  - if l==h then return f2(A[l])
  - m = (l+h)/2
  - return f1(A, l, m) + f1(A, m+1, h)
- Algorithm XXXX(A)
  - n = A.length
  - f1(A,1,n)

(a) Compute the output of the Algorithm when [123, 45, 52, 61] is given as an input. [2 marks]

(b) Describe the functionality of the Algorithm [2 marks]

(c) Is it possible to remove any lines in functions f1 or f2 such that the functionality of the algorithm does not change. If so, identify the lines. [2 marks]

(d) Compute the time complexity of the algorithm [2 marks]

(e) If the algorithm returns zero for an input array L, comment on the elements of L? [2 marks]

5. A common subsequence Z of two strings X and Y is said to be a 'Distinct Common Subsequence (DCS)' X & Y if Z has no symbols repeated in it. If $X = ABCDEF$, $Y = BADEAF$, then $ADEF$ is a DCS of X and Y. EF is also another DCS of X & Y. Given two strings X of length n and Y of length m, Design a pseudocode to compute all the distinct common subsequences of X and Y, of length greater than $\lfloor \frac{m+n}{2} \rfloor$. Here, $\lfloor . \rfloor$ is the usual floor function. If there are no DCS possible with the sepcified length, your pseudocode should return -1. Your design component should contain all the required steps. Analyse the algorithm with all the required steps. [15 marks]

6. Consider a grid of size $m \times n$, with m rows and n columns. Every cell is referred with a pair $[i, j]$ which conveys that the cell is at the intersection of the i^{th} row and the j^{th} column. You have to fill the cells of the grid with the numbers from the set $A = \{1, 2, ...9\}$. Given an $m \times n$ grid, design a pseudocode to fill the cells with the elements of A such that no row contains more than two even numbers, no column contains more than two odd numbers and the diagonal contains no two same numbers. Your pseudocode should return $m \times n$ array, R such that $R[i, j] = x$ if the cell $[i, j]$ contains the number x. [15 marks]

7. Given a graph $G = (V, E, s, c, t)$, where V is the set of vertices, E is the set of edges, $s \in V$ is a vertex designated as a source vertex, c is the set of capacities of all the edges of G, t is a vertex designated as the target vertex. Let $|f|$ denote the maximum flow of the network G with s as the source vertex and t as the target vertex. Given the graph $G = (V, E, s, c, t)$, design a pseudocode to identify the edges (u, v) for which $[c[(u, v)] - f[(u, v)]] \neq 0$, when the flow across the network is $|f|$. For the purpose of the illustration, you have to take a graph with a minimum of six vertices and a minimum of eight edges. Your design component should contain all the required steps. Analyze the pseudocode with all the required steps. [15 marks]

8. A string is said to be symmetrical if both halves of the string are the same. For example strings aa, abab, abaaba are a symmetrical string of length 2,4, and 6 respectively. Given a string S, develop an algorithm to determine the longest symmetrical substring of S. For example, given a string S1 = aababaacabacab, the longest symmetrical substring is acabacab which starts at index 7. [15 marks]

Final Assessment Test (FAT) - APRIL/MAY 2023

Programme	B.Tech	Semester	Winter Semester 2022-23
Course Title	DESIGN AND ANALYSIS OF ALGORITHMS	Course Code	BCSE204L
Faculty Name	Prof. Janaki Meena M	Slot	B1+TB1
		Class Nbr	CH2022235000280
Time	3 Hours	Max. Marks	100

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

Section A (4 X 10 Marks)
Answer All questions

01. Let $\Sigma = \{0, 1, 2, 3, \ldots 9\}$. We call 1 is the successor of 0, 2 is the successor of 1, ... and 9 is the successor of 8. A number in which all the digits from the left are in successive order, is called as an LR-Successive number (LRS-number). 789 is an LR-Successive number where as 798 is not an LR-Successive number. Similarly, a number in which all the digits from the right are in successive order, is called as an RL-Successive number (RLS- number). Two numbers are said to be digit-distinct number (dd-number) if there are no digits common between the numbers. 123 and 789 are dd-numbers. [10]

Given a set S that consists of either RLS-numbers or LRS-numbers, design a greedy-based pseudocode to compute $S' \subset S$, with the maximum number of dd-numbers. If $S = \{123, 234, 789, 4567, 98765, 89\}$. $\{123, 4567, 89\}$ shall be one of the subset of S, with maximum number of dd-numbers. Your design component should contain all the required steps. Analyse the algorithm with all the required steps.

Rubrics : Logic(2 marks), Illustration (3 marks), Pseudocode (3 marks), Running time \& Time-complexity (2 marks)

02. Consider the 2-dimensional plane where the points are represented by a pair of integers. A line with the end-points p_i and p_j is represented as $l(p_i, p_j)$. A polygon $P = \{p_1, p_2, \ldots, p_m\}$ is a set of points that forms a closed figure with the edges $\overline{p_1p_2}, \overline{p_2p_3}, \overline{p_3p_4}, \ldots, \overline{p_{(m-1)}p_m}, \overline{p_mp_1}$. Here $p_1, p_2, , , , . p_m$ are called the nodes of the polygon. Edges of the polygon can also be called as the line connecting the points. For example, the edge $\overline{p_1p_2}$ can be called as a line l whose end-points are p_1 and p_2. A line connecting the non-adjacent nodes of the polygon is called as the diagonal of the polygon. There will be more than one diagonal for a polygon. [10]

Given a polygon $P = \{p_1, p_2, \ldots, p_m\}$, design a pseudocode to identify the pair of diagonals of the P which intersect. Your design component should contain all the required steps. Analyze the pseudocode with all the required steps.

Rubrics : Logic(2 marks), Illustration (3 marks), Pseudocode (3 marks), Running time \& Time-complexity (2 marks)

03. Understand the Algorithm and answer the following questions . [10]

- Algorithm f(A, l, h, s) , A is an array such that A[i] $\geq$ A[i+1], $\forall$ i=1 to n

```
- m = (l+h)/2
- If m==A.length
-       If s < A[A.length]:
-             return A.length+1
-       else:
-             return A.length
- If m==1:
-       if s > A[1]:
-             return 1
-       else:
-             return 2
- If s < A[m-1] and s ≥ A[m]:
-             return m
- elseif s <A[m] and s ≥A[m+1]:
-       return m+1
- elseif s < A[m-1] and s < A[m]:
-       return f(A, m+1, h, s)
- elseif s > A[m+1] and s > A[m]:
-       return f(A, l, m-1, s)
```

a. Given an ordered array A = [31, 27, 24, 18, 14, 12], whose position-index starts with 1. Compute the output of the Algorithm f when s= 25, s=6, s=30, s= 33. You are required to give four outputs, one each for a value of s. [4 marks]

b. Describe the functionality of the above algorithm [2 marks]

c. For the array A (given in 4(a)), what shall be the value of s if the output is 4? [2 marks]

d. Compute the time complexity of the algorithm f. [2 marks]

04. a. Consider a problem Q: Given two positive integers N_1 and N_2, the task is to compute all the common digit(s) that occur(s) in N_1, N_2 and $(N_1 * N_2)$ (Here $*$ is the usual multiplication operation) . For example, given two numbers 21 and 12, your algorithm should return 2, since 2 is the digit that occurs in 21, 12 and (252 = 21*12). Identify the complexity-class (P/NP/NPC) of the problem Q with justification. [10]

b. Consider a problem P_1 : Given a set S of integers and an integer t, does there exist a subset S' of S such that the sum of the elements of S' equals t? For example, for the input $S = \{3, 13, 24, 45, 102\}$ and $t = 72$, the solution to the problem is 'yes' since there exists the subset $S' = \{3, 24, 45\}$ whose sum of the elements is 72. Identify the complexity-class (P/NP/NPC) of the problem P_1 with justification.

Rubrics: Identification of class-complexity (2 marks), Justification (3 marks)

Section B (4 X 15 Marks)
Answer All questions

05. $A_1 * A_2 * A_3 * \ldots * A_n$ is called as a multiplicative chain of length n, where $A_1, A_2, \ldots, A_n$ are matrices. For the three matrices A_1, A_2, A_3,, multiplicative chains are $A_1 * A_2 * A_3$, $A_1 * A_3 * A_2$, $A_2 * A_3 * A_1$, $A_2 * A_1 * A_3$, $A_3 * A_1 * A_2$, $A_3 * A_2 * A_1$. If there is a matrix A such that $A = A_1 * A_2 * A_3 * \ldots * A_n$, then A is called the value of the multiplicative chain. A multiplicative chain of length n is called a Feasible Multiplicative Chain (FMC) if there is a value for that chain. Given three matrices A_1, A_2, A_3 with the respective sizes $(2,3), (4,2), (3,5)$, then the multiplicative chain $A_2 * A_1 * A_3$ is an FMC . $A_1 * A_2 * A_3$ is not an FMC since there is no value for $A_1 * A_2 * A_3$. [15]

Given n matrices $A_1, A_2, A_3, \ldots, A_n$ with the respective sizes $(r_1, c_1), (r_2, c_2), \ldots, (r_n, c_n)$, design a dynamic programming based pseudocode to compute all the FMCs and identify the FMC that requires minimum number of scalar multiplication among all the FMCs. If there are two FMCs with same number of minimum multiplications, your pseudocode should return both. For the given matrices, if no FMC is possible, your pseudocode should return 0. Your design component should contain all the required steps. Analyse the algorithm with all the required steps.

Rubrics: Logic(3 marks), Illustration (5 marks), Pseudocode (5 marks), Running time \& Time-complexity (2 marks)

06. A string Y is said to be an 1-order cyclic rotation of a string X if Y is obtained from X by the process of shifting the last character of X to the first position and shifting all the other characters of X one position to the right. Similarly we define the cyclic rotation of 2-order, 3-order and so on. For example, the 1-order cyclic rotation of the string *abcd* is *dabc*, 2-order cyclic rotation of the string *abcd* is *cdab*, 3-order cyclic rotation of the string *abcd* is *bcda*. [15]

Given two strings S_1 and $S2$, design a brute-force algorithm A to check if $S2$ is a 1-order cyclic rotation of S_1 or vice versa. Also design a string-matching algorithm B (other than Brute-Force) to compute whether S_2 is a 1-order cyclic rotation of S_1 or vice-versa. Your design component should contain all the required steps. Analyze the pseudocode with all the required steps.

Rubrics: Logic(3 marks), Illustration (5 marks), Pseudocode (5 marks), Running time \& Time-complexity (2 marks)

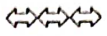
07. Given a graph $G = (V, E, s, c, t)$, where V is the set of vertices, E is the set of edges, $s \in V$ is a vertex designated as a source vertex, c is the set of capacities of all the edges of G, t is a vertex designated as the target vertex. Let $|f|$ denote the maximum flow of the network G with s as the source vertex and t as the target vertex. Given the graph $G = (V, E, s, c, t)$, design a pseudocode to identify the edge(s) (u, v) which contributes more to $|f|$. For the purpose of the illustration, you have to take a graph with a minimum of six vertices and a minimum of eight edges. Your design component should contain all the required steps. Analyze the pseudocode with all the required steps. [15]

Rubrics: Logic(3 marks), Illustration (5 marks), Pseudocode (5 marks), Running time \& Time-complexity (2 marks)

08. Consider a grid of size $m \times n$, with m rows and n columns. Every cell is referred with a pair $[i, j]$ which conveys that the cell is at the intersection of the i^{th} row and the j^{th} column. Column-adjacent cells of the cell $[i, j]$ are the cells $[i-1, j]$, $[i+1, j]$. Row-adjacent cells of the cell $[i, j]$ are the cells $[i, j-1]$, $[i, j+1]$. Diagonal-adjacent cells of $[i, j]$ are $[i-1, j-1]$, $[i-1, j+1]$, $[i+1, j-1]$, $[i+1, j+1]$. Two cells are said to be adjacent if the $k \geq 5$ cells are either row-adjacent or column-adjacent or diagonal-adjacent. Given k colours [15]

and an $m \times n$ grid, design a backtracking pseudocode to colour the cells of $,C_1, C_2, C_3, \ldots C_k$ the grid such that no two adjacent cells are coloured same. Your pseudocode should return $m \times n$ array, R such that $R[i, j] = C_j$ if the cell $[i, j]$ is coloured with C_j. Your design component should contain all the required steps. Analyze the pseudocode with all the required steps

Rubrics : Logic for A and B)4 marks(, Illustration for A and B)4 marks(, Pseudocodes : A and B)4 marks(, Running time \& Time-complexity of A and B)3 mark.(



VIT

Vellore Institute of Technology

Final Assessment Test (FAT) – May 2022

Programme	B.Tech	Semester	Winter Semester 2021-22
Course Title	DESIGN AND ANALYSIS OF ALGORITHMS	Course Code	CSE2012
Faculty Name	Prof. S Venkatraman	Slot	C2+TC2
		Class Nbr	CH2021225000708
Time	3 Hours	Max. Marks	100

Answer all the questions.

If any assumptions are required, assume the same and mention that assumption in the answer script.

Use of intelligence is highly appreciated.

Every question has 'design' and 'analysis' component.

Design component of your answer should include : logic, pseudocode and an illustration on the functionality of the pseudocode

Analysis component of your answer should include: computation of the running time and the time complexity.

Every question (other than question nos:5,9) carries 10 marks. Marks for the different components are : description of logic/technique(2 marks), pseudocode(3 marks), illustration (3 marks), running time (1 mark) and time-complexity(1 mark)

Marks for the different components of question no.5(a), 5(b) : Computation of the complexity class(2 marks), Justification(3 marks)

Marks for the different components of question no.9 : Problem formulation(2 marks), pseudocode(3 marks), illustration(3 marks), running-time(1 mark), time-complexity (1 mark)

Section-1 (10 X 10 Marks)

Answer All questions

1. A Swap-operation of three numbers a_1, a_2, a_3 in an n-digit number $N = a_1 a_2 \ldots a_3 \ldots a_4 \ldots$ written as $Swap(a_1, a_3, a_4)$ is defined as follows: [10]

   - a_1 takes the position of a_3 in N.
   - a_3 takes the position of a_4 in N.
   - a_4 takes the position of a_1 in N.

   $Swap(1, 5, 7)$ on 1267 gives a new number 7215. Every swap-operation of three numbers on N gives a new number. Given a number N, design an algorithm to compute the triplet a_1, a_3, a_4 such that number M obtained from N through the $Swap(a_1, a_3, a_4)$ is the maximum among all the numbers obtainable from N through the swap of three numbers. Analyse your algorithm with the running-time and the time-complexity.

2. Given an array A of n integers, design an algorithm to arrange the elements of N in such a way that $A[i] \leq A[i+1]$ and $|A[i] - A[i+1]|$ is always less than k, for any $1 \leq i \leq n-1$. In other words, the successive elements should be in an increasing order and the difference between the successive elements should be less than k. If any element of A could not be arranged satisfying the above specified conditions, your algorithm should delete those numbers from the array. For example, given the array $< 1, 7, 2, 3, 9 >$ and $k = 2$, your algorithm should output $< 1, 2, 3 >$. Analyse your algorithm with the running-time and the time-complexity. [10]

3. The convex hull of a set X of points is the smallest convex polygon P for which each point in X is either on the boundary of P or in its interior. Let P and Q be two convex hulls in a two-dimensional plane covering the set X and the set Y respectively where $P = \{p_1, p_2, \ldots, p_m\}$ and $Q = \{q_1, q_2, \ldots, q_n\}$. Here the points p_i, $1 \leq i \leq m$ are the vertices of the convex hull P. Similarly, points q_i, $1 \leq i \leq n$ are the vertices of the convex hull Q. Two Polygons are said to intersect if there is at least one point (either boundary point or interior point) common between the two polygons. Given the points $p_1, p_2, \ldots, p_m, q_1, q_2, \ldots, q_n$, design an algorithm to check whether P and Q intersect or not?. Here, n and m are any two positive integers. Analyse your algorithm with the running-time and the time-complexity. [10]

First-to-last-cycle is a function that operates on an array S, written as $f(S)$, described as follows: $f(S[1,2,...,m]) = S'[1,2,...,m]$ where $S'[i] = S[i+1]$, where $0 < i < m$, $S'[m] = S[1]$. For example, $f(abcd) = bcda$ $f^2(abcd) = f(f(abcd)) = f(bcda) = cdab$ $f^3(abcd) = f(f(f(abcd))) = dabc$. Text T is an array $T[1,2,3...,n]$ of length n and the pattern P is an array $P[1,2,...,m]$ of length m. We define, $Shift(T,P) = s$, if $P[1,2,...,m] = P[1,2,...,m]$ where $0 \leq s \leq n$ and $Shift(T,P) = -1$ if pattern P does not occur in text T. $Shift(T,P)$ may return one integer or a sequence of integers. Given a Text T and a Pattern P, design an algorithm to compute $Shift(T, f^k(P))$, where $k = 0,1,2,3,...m$. Note that $f^0(P) = P$. For example, if $T = abcbcbcacabcacbcbc$ and $P = abc$ then $Shift(T, f^0(P) = 0, 16$, $Shift(T, f(P)) = 5$, $Shift(T, f^2(P)) = 8$. Analyse your algorithm with the running-time and the time-complexity.

5. (a) A word $a_1a_2a_3...a_n$ is said to be a palindrome if $a_1 = a_n$, $a_2 = a_{n-1}$ and so on. The word 'madam' is a palindrome and the 'eat' is not a palindrome. Given a word w, the task of the 'palindrome-check problem' is to decide whether the given word w is a palindrome or not. With proper justification, Compute the complexity class (P class or NP class or NP complete class) of 'palindrome-check problem'. Also check whether the problem falls into more than one complexity class and if so, justify your answer. [10]

(b) Let A be an array which contains n numbers. The task of the 'frequency-count problem' is to identify the numbers which occur one time, numbers which occur two times, numbers which occur three times and so on and the numbers which occur n times. With proper justification, Compute the complexity class of 'frequency-count problem. Also check whether the problem falls into more than one complexity class and if so, justify your answer.

6. Given a flow network $G = (V, E, c, s, t)$ where V is the set of vertices , E is the set of edges connecting two vertices of V, c is the set of capacities, s is in V designated as source vertex and t is in V designated as target vertex. Every edge $(u,v) \in E$ has a positive capacity. An edge e in G is upper-binding if increasing e's capacity by 1 increases the value of the maximum flow in G. Similarly, an edge is lower-binding if decreasing its capacity by 1 decreases the value of the maximum flow in G. Given a flow network G with the capacities, design an algorithm to identify all the upper binding edges and all the lower binding edges of G. Analyse your algorithm with the running-time and the time-complexity. [10]

7. Given a graph $G = (V, E)$, V is the set of vertices and E is the set of edges. A simple path in a graph is a path without repeated vertices. Let any two vertices in V be designated as Source S and Terminal T. A path, say, $S - v1 - v2 - T$ is said to be a path with 2 vertices. That is, for the calculation of the number of vertices in a path, we exclude the source and the destination of the path. Given the graph G, vertex S, vertex T and an integer k, design an algorithm to check whether there exist a simple path from S to T with at least k vertices and return the sequence of vertices that form the simple path with at least k vertices. Analyse your algorithm with the running-time and the time-complexity. [10]

8. An $n \times n$ grid has n^2 cells. A diagonal of the $n \times n$ grid that starts from the top-most right corner (right corner of the grid is the one which is to your right when you face the grid) of the grid and ends at the left-most bottom corner of the grid is called as anti-diagonal of the grid. Given an $n \times n$ power grid, the problem is to place n power stations in n cells in such a way that the following conditions are staisfied. (i). Any column of the grid cannot have more than one power station. (ii). Any row of the grid cannot have more than one power station. (iii). Anti-diagonal cannot have more than one power station. Given n, design an algorithm that will return the position of the cells where n power stations can be placed in the given grid. Analyse your algorithm with the running-time and the time-complexity. [10]

9. Propose a problem P (of your choice) in detail which can be solved by a dynamic programming based algorithm, with justification. You are not supposed to describe any problem described in this question paper. Design an algorithm for the problem chosen by you. Analyse your algorithm with the running-time and the time-complexity. [10]

10. You are organizing a function for which you have invited n guests. Every guest will be picked up from their house and dropped at the venue of the function. Every car will start from the venue of the function to pick up the guest and drop the guest at the venue. Every guest g_i will have a pair (s_i, d_i) where s_i is the start-time of the car from the venue to pick the guest and the d_i is the drop-time of the guest g_i at the venue. Given the details (s_i, d_i) of the guests g_i, $i = 1,2,...,n$, design an algorithm to compute the minimum number of cars to be booked for the purpose. For example, If $g_1 : (8:15, 9:05), g_2 : (8:40, 9:25), g_3 : (9:10, 9:45), g_4 : (9:47, 10:50), g_5 : (9:30, 10:20)$ then minimum of two cars are required. Analyse your algorithm with the running-time and the time-complexity. [10]

Final Assessment Test (FAT) – November/December 2022

Programme	M.Tech. (Integrated)	Semester	Fall Semester 2022-23
Course Title	DESIGN AND ANALYSIS OF ALGORITHMS	Course Code	CSE3037
Faculty Name	Prof. Smrithy G S	Slot	A1+TA1
		Class Nbr	CH2022231000983
Time	3 Hours	Max. Marks	100

Section A (10 X 10 Marks)
Answer All questions

1. What is a binomial heap?. State the properties of binomial heaps. **(2 Marks)** **[10]**
   Insert the elements 7, 2, 4, 17, 1, 11, 6, 8, 15, 10, 20, 5 one by one into an empty minimum binomial heap. **(4 Marks)**
   Delete the elements 1, 2, 4, 15 one by one from the above constructed binomial heap. **(4 Marks)**
2. State the properties of Red Black tree. **(2 Marks)** **[10]**
   Insert the elements 10, 18, 7, 15, 16, 30, 25, 40, 60 one by one into an empty red black tree. **(4 Marks)**
   Delete the elements 16, 7, 10, 30, 18 one by one from the above constructed Red Black tree. **(4 Marks)**
3. Given two strings, design an algorithm to determine longest common subsequences (LCS) using dynamic programming approach. **(4 Marks)** **[10]**
   Illustrate the algorithm developed for the strings S1= BDCABA, S2= ABCBDAB. Also print all the possible LCS string sequences. **(6 Marks)**
4. Given two positive integers x and n, write a function to compute x^n. For example, if x=2 and n=4 then $x^n=16$. **[10]**
   Design a brute force algorithm to solve the given problem and analyse the time complexity. **(5 Marks)**
   Design a Divide and Conquer algorithm to solve the same problem and analyse the time complexity. **(5 Marks)**
5. Given a 3×3 board with 8 tiles (every tile has one number from 1 to 8) and one empty space. The objective is to place the numbers on tiles to match the final configuration using the empty space. You can slide four adjacent (left, right, above, and below) tiles into the empty space. **[10]**
   Consider the following initial and final configuration,

   Initial configuration

2	8	3
1	6	4
7		5

   Final configuration

1	2	3
8		4
7	6	5

Apply least cost branch and bound strategy and find the minimum number of moves required to reach the final configuration from the initial configuration. **(10 Marks)**

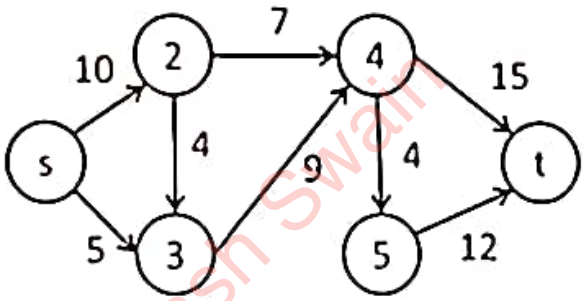
6. a. State KMP pattern matching algorithm. Find the shift for the pattern P: "ABAABABA" in the given text T: "ABAABCABAABABAE" using KMP string pattern matching algorithm **(6 Marks)** **[10]**

   b. How does KMP algorithm differs from naive string matching algorithm. Analyse the best and worst case time complexity of both algorithms. **(4 Marks)**

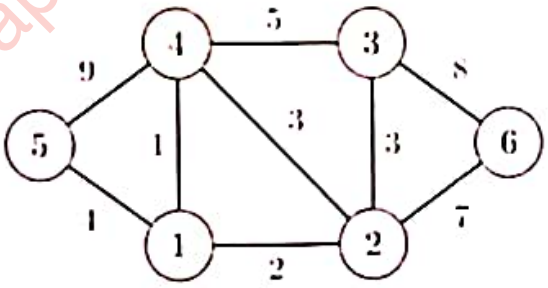
7. a. Given two line segments (A, B) and (C, D) design an algorithm to find whether the given line segments intersect with each other or not. **(6 Marks)** **[10]**

   b. Illustrate your algorithm for various intersecting and non-intersecting line segments. **(4 Marks)**

8. Find maximum flow using Ford-Fulkerson algorithm for the given graph. For each iteration, specify the flow network, residual network and selected augmenting path and its capacity. **(10 Marks)** **[10]**



9. Determine minimum spanning tree for the given graph using Prim's algorithm. Illustrate step by step procedure and analyse the time complexity of the algorithm. **(10 Marks)** **[10]**



10. What is reducibility? State 3-SAT and Clique problem. Reduce a 3-SAT problem into Clique problem with an example. **(10 marks)** **[10]**

Reg. No: 22BCE0597

F/E/TX

Final Assessment Test – May 2024

Course: BCSE204L - Design and Analysis of Algorithms

Class NBR(s): 0765/0776/0817/0825/0829/0831/0839 /0874/0877/0884/0890/0901/0906/0912/0917/0923 /0925/0927/0929/0931/0933/0939/6134 Slot: A1+TA1

Time: Three Hours Max. Marks: 100

- KEEPING MOBILE PHONE/ELECTRONIC DEVICES EVEN IN 'OFF' POSITION IS TREATED AS EXAM MALPRACTICE
- DON'T WRITE ANYTHING ON THE QUESTION PAPER

Answer any TEN Questions

(10 X 10 = 100 Marks)

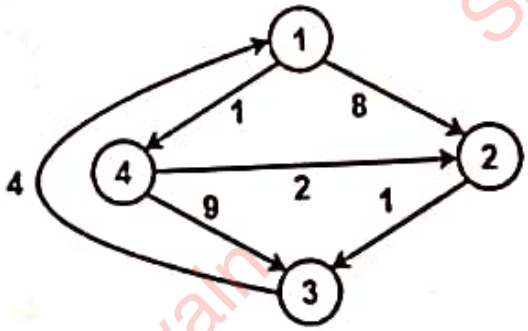
1. Design an algorithm for sum of all elements of a matrix using 2D Array and analyze its time complexity. How to prove the correctness of matrix using Induction method.

2. Outline important factors of Dynamic Programming and perform Matrix chain multiplication for {4, 10, 3, 12, 20, and 7). The matrices have size 4 x 10, 10 x 3, 3 x 12, 12 x 20, 20 x 7. Compute M [i,j], 0 ≤ i, j≤ 5. Consider M [i, i] = 0 for all i.

3. a) Interpret steps of KMP_next array and the Knuth-Morris-Pratt (KMP) algorithm. Compute the KMP_next array of the Text as ABCCDDAEFG and pattern as CDD. [5]

   b) Classify between KMP and Rabin-karp algorithm works with strength and weakness of Complexity. [5]

4. Perform step-by-step process of find the shortest path distance between every pair of vertices using Floyd warshall algorithm, from the directed weighted graph.



5. Relate Graham's Scan and Jarvis March Algorithm steps and how to choose the best practices for computational Convex Hull. Illustrate with an example.

6. Narrate how Randomized Quick Sort algorithm differs from Normal Quick Sort. Write an algorithm and solve the randomized quick sort algorithm with an example.

7. Compare and Contrast the following terms used in computational complexity such as N, NP and NP-complete problems. Discuss its features with real time examples.

VIT
Vellore Institute of Technology

Final Assessment Test – May 2024

Course: BCSE204L - Design and Analysis of Algorithms

Class NBR(s): 0772/0796/0803/0822/0827/0834/ 0876/0879/0887/0891/0902/0909/0914/0919/ 0924/0926/0928/0930/0932/0934/0935/0937/ 0941/6646

Slot: A2+TA2

Time: Three Hours

Max. Marks: 100

- KEEPING MOBILE PHONE/ELECTRONIC DEVICES EVEN IN 'OFF' POSITION IS TREATED AS EXAM MALPRACTICE
- DON'T WRITE ANYTHING ON THE QUESTION PAPER

Answer any TEN Questions

(10 X 10 = 100 Marks)

1. a) Solve the recurrence $T(n) = 4T(\sqrt{n}) + log_2^5 n$ [4]

   b) Write an iterative algorithm to find the sum $\sum_{i=2}^{n} \frac{1}{i(i-1)}$, $n \in Z$. Prove that your algorithm is correct. [6]

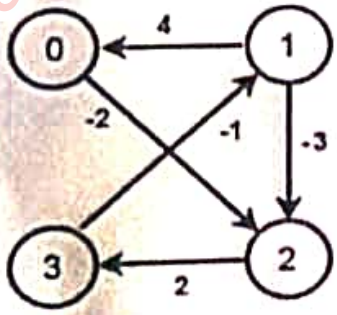
2. You are given two DNA sequences, DNA1 and DNA2, consisting of characters from the alphabet {'A', 'C', 'G', 'T'}. Your task is to find the minimum number of single-character edits (insertions, deletions, or substitutions) required to transform DNA1 into DNA2. For example, given DNA1 = "AGTACG" and DNA2 = "GTCAGT", the minimum number of edits required is 4. One possible sequence of edits results in transforming DNA1 into DNA2 with 4 edits is as follows:

   1. Substitute 'A' in DNA1 with 'G': "GGTACG"
   2. Substitute 'G' in DNA1 with 'T': "GTTACG"
   3. Delete 'A' from DNA1: "GTTCG"
   4. Substitute 'C' in DNA1 with 'A': "GTTAG"

   Develop a DP algorithm that will provide an optimized solution to determine the minimum edit distance between the two DNA sequences.

3. Assume that you have a text T and a pattern P. Write an algorithm to report the index of the first occurrence (if any) of P in T from the π-table of PT where PT is the concatenation of P and T. If P does not occur in T, then print -1. Trace the working of your algorithm with the help of a positive example.

4. Given a directed weighted graph G(V, E) with potentially negative weight edges, check whether the implementation of Bellman-Ford algorithm can find the shortest path between the nodes. How will your algorithm determine the possibility of negative cycles in the graph? Explain the reasoning behind the chosen approach.



5. Write the pseudocode for Jarvis-march algorithm and trace it for the points provided in the diagram below.

VIT

Reg. No. : 21BPS12250

Final Assessment Test (FAT) - July/August 2023

Programme	B.Tech.	Semester	Fall Inter Semester 22-23
Course Title	DESIGN AND ANALYSIS OF ALGORITHMS	Course Code	BCSE204L
Faculty Name	Prof. Dr.Rajakumar Arul	Slot	A1+TC1
		Class Nbr	CH2022232500925
Time	3 Hours	Max. Marks	100

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

Section A (4 X 10 Marks)
Answer all questions

01. Given a set of n points $\{C_1(x_1, y_1), C_2(x_2, y_2), \ldots, C_n(x_n, y_n)\}$ that represents cities $C_1, C_2 \ldots C_n$, where the x-coordinate represents the longitude and the y-coordinate represents the latitude of the respective city, design an algorithm using the Divide-Conquer-Combine strategy (*DCC*) to arrange the cities in the decreasing order of longitude. For the purpose of this problem, assume that the latitudes and longitudes are positive integers without involving any directions. When two cities have the same longitude arrange those cities based on the decreasing order of latitude. [10]

Rubrics:

Logic(2 marks), Illustration (3 marks), Pseudocode (3 marks), Running time & Time-complexity (2 marks)

02. Understand the following algorithm and answer the questions below: [10]

Algorithm 1 : $F(A)$

```
0: Input : A is an array of positive integers
1: Here '/' performs integer division
2: n = A.length()
3: for i = 1 to n do
4:     A[i] = F1(A[i])
5: end for
6: return A
7:
8: Algorithm : F1(n1)
9: s1 = 0 ;
10: s1 = s1*10 + n1 mod 10 ;
11: n1 = n1 / 10 ;
12: s1 = s1*10 + n1 mod 10 ;
13: n1 = n1 / 10 ;
14: n1 = n1 * 100 ;
15: n1 = n1 + s1 ;
16: return n1
```

(i) What will be the output of the algorithm F when the input array A = [123, 4578, 2391, 4165]? [2 Marks]

Final Assessment Test (FAT) - May 2024

Programme	B.Tech.	Semester	WINTER SEMESTER 2023 - 24
Course Title	DESIGN AND ANALYSIS OF ALGORITHMS	Course Code	BCSE204L
Faculty Name	Prof. Jayaram B	Slot	A1+TA1
		Class Nbr	CH2023240502391
Time	3 Hours	Max. Marks	100

General Instructions:

- Write only Register Number in the Question Paper where space is provided (right-side at the top) & do not write any other details.
- 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: Computation of T(n), Time-complexity.

Section - I

Answer all questions (4 X 10 Marks = 40 Marks)

01. Let n be a positive integer. Let $\Sigma_1 = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, \ldots n^2\}$. $\Sigma_2 = \{a, b, c, \ldots, z\}$. Consider a grid of size $n \times n$, with n rows and n columns **[10]**

Given a grid of size $n \times n$, design a backtracking-based pseudocode to fill the cells of the grid with elements from $\Sigma_1 \cup \Sigma_2$ such that (i) all rows start and end with elements from Σ_2 (ii) all columns start and end with elements from Σ_1 (iii) Condition (i) and Condition (ii) is exempted for the first row, last row, first column and the last column of the grid (iv) No two rows, No two columns, No two diagonals have any repetition of elements from $\Sigma_1 \cup \Sigma_2$. Your design component should contain all the required steps and the analysis component should contain all the required components.

[**Rubrics:** Logic(2 marks), Illustration (3 marks), Pseudocode (3 marks), Running time & Time-complexity (2 marks)]

02. Given a graph $G = (V, E, s, c, t)$, where V is the set of vertices, E is the set of edges, $s \in V$ is a vertex designated as a source vertex, c is the set of capacities of all the edges of G, t is a vertex designated as the target vertex. Let $|f|$ denote the maximum flow of the network G with s as the source vertex and t as the target vertex. **Given the graph $G = (V, E, s, c, t)$, design a pseudocode to compute the maximum flow from s to t such that the total number of** **[10]**

augmenting paths computed in the pseudocode to calculate $|f|$, should not be greater than the number of edges in the graph. Your design component should contain all the required steps. Analyze the pseudocode with all the required steps.

[**Rubrics**: Logic(2 marks), Illustration (3 marks), Pseudocode (3 marks), Running time & Time-complexity (2 marks)]

03. a) Consider the problem P: Given a positive integer n, the task is to compute $n^{\log_{10} n}$. Compute the class-complexity of the problem P (P/NP/NPC) with justification. [5 marks] **[10]**
[**Rubrics** : Identification of class-complexity (2 marks), Justification (3 marks)

b) Consider the problem, **NAVIGATION** : Consider an n- digit number $a_1a_2a_3 \ldots a_n$, with distinct digits. Assume that $a_1, a_2, \ldots, a_n$ form the nodes of an undirected graph G. There will be an edge connecting a_i and a_j in G iff $i < j$. Task is to navigate from a node in G and reach the same node in such a way that, every node is visited only once except the starting and the ending node. As an example, for the number is12347, one possible NAVIGATION path is:
$1 - 2 - 3 - 4 - 7 - 1$.
Compute the class-complexity of the problem **NAVIGATION** (P/NP/NPC) with justification. [5 marks]

04. Let $\Sigma = \{a, b, c, \ldots x, y, z\}$. Let T and T_1 be any two strings with symbols from Σ. T is said to be a cousin of T_1 if T_1 can be obtained from T by shifting all the symbols(except the first symbol) of T one position to left and the first symbol of T is shifted to the rightmost position, or viceversa. The string $abcdef$ is a cousin of $bcdefa$. **Given two strings T, T_1, design a pseudocode to decide whether T and T_1 are cousins or not. Your pseudocode should use any of the string-matching pseudocodes discussed in the class.** Your design component should contain all the required steps. Analyze the pseudocode with all the required steps. [10 marks] **[10]**

[**Rubrics**: Logic(2 marks), Illustration (3 marks), Pseudocode (3 marks), Running time & Time-complexity (2 marks)]

Section - II

Answer all questions (4 X 15 Marks = 60 Marks)

05. Consider the 2-dimensional plane where the points are represented by a pair of integers. An edge with the end-points p_i and p_j is a line represented as $l(p_i, p_j)$, connecting the points p_i and p_j. A polygon $P = \{p_1, p_2, \ldots, p_m\}$ is a set of points that form a closed figure with the edges $p_1 - p_2, p_2 - p_3, p_3 - p_4, \ldots, p_{(m-1)} - p_m, p_m - p_1$. The points $p_1, p_2, \ldots p_m$ are the vertices of the polygon. A polygon is said to be simple if none of the edges of the polygon cross itself. A convex polygon is a simple polygon where, any line segment connecting two points within the polygon will lie entirely inside the polygon or on its boundary. Convex Hull of a set of points is the smallest simple convex polygon that contains all the points in the set. Circumcircle of a convex hull of the points $p_1, p_2, \ldots, p_n$, is the circle that passes through all the vertices of the convex hull. **[15]**
Given n points $\{p_1, p_2, \ldots, p_n\}$ in the 2-dimensional plane, design a pseudocode to compute the radius of the circumcircle of the convex hull of the given n points. Your 'design' component should contain all the required steps and the 'analysis' component should contain all the required steps. [15 marks]

[**Rubrics**: Logic(4 marks), Illustration (4 marks), Pseudocode (4 marks), Running time & Time-complexity (3 marks)]

06. You are organizing a function for which you have invited n guests. Every guest will be picked up from their house and dropped at the venue of the function. **[15]**
Every car will start from the venue of the function to pick up the guest and drop the guest at the venue. Every guest g_i will have a pair (s_i, d_i), where s_i is the start-time of the car from the venue to pick the guest and the d_i is the drop-time of the guest g_i at the venue. **Given the details (s_i, d_i) of the guests $g_i, i = 1, 2..., n$, design an algorithm to compute the minimum number of cars to be booked for the purpose.**
For example, If
$g_1 : (8:15, 9:05), g_2 : (8:40, 9:25), g_3 : (9:10, 9:45), g_4 : (9:47, 10:50), g_5 : (9:20, 10:20)$
then minimum of two cars are required. Your `design' component should contain all the required steps and the `analysis' component should contain all the required steps. [15 marks]

[**Rubrics**: Logic(4 marks), Illustration (4 marks), Pseudocode (4 marks), Running time & Time-complexity (3 marks)]

07. Given a chain of n matrices $< A_1, A_2, \ldots, A_n >$ where each matrix A_i is of size $p_{(i-1)} \times p_i, i = 1, 2, 3, \ldots n$. Product of the chain of matrices $< A_1, A_2, \ldots, A_n >$ is the value of the expression $A_1 \times A_2 \times A_3 \ldots \times A_{n-1} \times A_n$. Given an array $[p_0, p_1, p_2, \ldots, p_n]$, **Design a divide-conquer-combine pseudocode to parenthesize the chain $< A_1, A_2, \ldots, A_n >$ such that minimum number of scalar multiplications are involved in the computation of the product of the chain.** Your `design' component should contain all the required steps and the `analysis' component should contain all the required components. [15 marks] **[15]**
[**Rubrics**: Logic(4 marks), Illustration (4 marks), Pseudocode (4 marks), Running time & Time-complexity (3 marks)]

08. Given an unlimited supply of coins of denominations $x_1, x_2, \ldots, x_n$ and a value v, the task is to make change for the value v. That is, for the given v, we have to find a set of coins whose total value is v. If the denominations are 5 and 10, we can make the change for 15 as 10+5 and we can not make the change for 12. **Given the denominations $x_1, x_2, \ldots, x_n$ and a value v, design an dynamic programming based algorithm to express v using denominations $x_1, x_2, \ldots, x_n$ such that minimum number of coins are required .** In case, if v can not be expressed in terms of the given denominations, your algorithm should return -1. Your `design' component should contain all the required steps and the `analysis' component should contain all the required components. [15 marks] **[15]**
[**Rubrics**: Logic(4 marks), Illustration (4 marks), Pseudocode (4 marks), Running time & Time-complexity (3 marks)]

VIT®
Vellore Institute of Technology

Reg. No. : 22BPS1059

Final Assessment Test(FAT) - Nov/Dec 2024

Programme	B.Tech.	Semester	Fall Semester 2024-25
Course Code	BCSE204L	Faculty Name	Prof. Jeipratha P N
Course Title	Design and Analysis of Algorithms	Slot	A1+TA1
		Class Nbr	CH2024250100543
Time	3 hours	Max. Marks	100

General Instructions

- Write only Register Number in the Question Paper where space is provided (right-side at the top) & do not write any other details.
- Answer all the EIGHT 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: Computation of T(n), Time-complexity.

Course Outcomes

1. Apply the mathematical tools to analyze and derive the running time of the algorithms
2. Demonstrate the major algorithm design paradigms.
3. Explain major graph algorithms, string matching and geometric algorithms along with their analysis.
5. Explain the hardness of real-world problems with respect to algorithmic efficiency and learning to cope with it.

Section - I

Answer all Questions (4 × 10 Marks) *M - Marks

Q.No	Question	*M	CO	BL
01.	Let $S = \{d_1, d_2, d_3, \ldots d_n\}$, where d_i's are integers of varying values. Concatenation of two integers, denoted as $d_1 \times d_2$, is defined as combining d_1 and d_2 in the same order and the cost of concatenation is $\lvert d_1\rvert \times \lvert d_2\rvert \times (\lvert d_1\rvert - \lvert d_2\rvert)$, where $\lvert d_1\rvert$ and $\lvert d_2\rvert$ represent the number of digits in d_1 and d_2 respectively. 123×2345 is 1232345. Cost of concatenation of $(123 \times 4567) \times 333$ is $4 \times 3 \times (-1) + 7 \times 3 \times 4 = 72$. Cost of concatenation of $123 \times (4567 \times 333)$ is $4 \times 3 \times 1 + 7 \times 3 \times (-4) = -72$. **Given a set S, Design a pseudocode to compute the parenthesization of $d_1 \times d_2 \ldots \times d_n$ in such a way that the cost of concatenation is minimum.** You have to illustrate your pseudocode by considering a set S with a minimum of eight elements. You are not supposed to use any illustration given in this question as part of your illustration in the answer. [**Rubrics:** Logic for pseudocode: 2 marks, Illustration for pseudocode : 4 marks, Pseudocode : 3 marks, Time-complexity :1 mark]	10	2	3

02.	**Consider a string W of length n, design a brute-force pseudocode to compute the maximal repeated substring that can appear in multiple places, but with no overlapping occurrences.** If $W = banana$, your pseudocode should return an and na. You have to illustrate your pseudocode by considering a string W of length at least 7 and there should be at least two maximal repeated substrings. You are not supposed to use any illustration given in this question as part of your illustration in the answer. [**Rubrics:** Logic for pseudocode: 2 marks, Illustration for pseudocode : 4 marks, Pseudocode : 3 marks, Time-complexity :1 mark]	10	2	3
03.	Consider the problem P: Given a positive integer N, the task is to compute all the even factors of N. For example, 2 is an even factor of 6. **Compute the class-complexity of the problem P (P/NP/NPC) with justification.** [5 marks] [**Rubrics:** Identification of class-complexity (2 marks), Justification (3 marks)] Consider the problem P_1: Given the set $S = \{1, 2, 3, \ldots, n\}$ and $X = \{s_1, s_2, \ldots, s_m\}$, s_i's are the subsets of S, the task is to identify the smallest subset of X whose union is S. For example, given $S = \{s_1 = \{1, 2, 3\}, s_2 = \{2, 4\}, s_3 = \{3, 4\}, s_4 = \{4, 5\}\}$, the smallest subset of X whose union is S is $\{s_1, s_4\}$. **Compute the class-complexity of the problem P (P/NP/NPC) with justification.** [5 marks] [**Rubrics:** Identification of class-complexity (2 marks), Justification (3 marks)]	10	5	4
04.	Consider an one-dimensional line. Consider the set of line segments, $S = \{l_1 = (a_1, b_1), l_2 = (a_2, b_2), \ldots, l_n = (a_n, b_n)\}$, where $l_n = (a_n, b_n)$ means that a_n is the starting point of the line segment l_n and b_n is the end point of the line segment l_n. **Given S, design a pseudocode to compute the maximum length of the line segment which overlaps with all the line segments of S.** Your algorithm should return the maximum length of the required line segment and the starting and the end point of that line segment. If there is no intersection among the segments, your pseudocode should return zero. For $S = \{[1, 4][2, 5][3, 6][7, 8]\}$, your pseudocode should output 1 and [3,4]. You have to illustrate your pseudocode by considering a set S with a minimum of 8 elements and maximum length of the required line segment should atleast be 2. You are not supposed to use any illustration given in this question as part of your illustration in the answer. [**Rubrics:** Logic for pseudocode: 2 marks, Illustration for pseudocode : 4 marks, Pseudocode : 3 marks, Time-complexity :1 mark]	10	3	3

Section - II

Answer all Questions (4 × 15 Marks) *M - Marks

Q.No	Question	*M	CO	BL
05.	**Given a string S, design two different pseudocodes: Pseudocode 1 , Pseudocode 2, such that the pseudocode 1 applies the divide-conquer-combine strategy, Pseudocode 2 applies a strategy different from divide-conquer-combine, to compute the longest substring of S which is also a palindrome (palindrome is a string that reads the same when read from left to right or vice versa).** For the input S= 'abbadefggfhhfgggg', your pseudocode should output 'ggfhhfgg'. For the illustration of your pseudocode, you are supposed to take a string S which has a minimum length of 10 and the length of the longest substring which is a palindrome should be atleast 3. [**Rubrics:** Logic for pseudocode: 2 marks, Illustration for Pseudocode 1 : 2 marks, Illustration for the Pseudocode 2 : 3 marks, Pseudocode 1 : 3 marks, pseudocode 2 : 2 marks, Time-complexity of the pseudocode 1 : 2 marks, Time-complexity of the pseudocode 2 : 1 mark]	15	2	6

06.	Given an $N \times M$ grid matrix, the cells of the grid may contain either english vowels or english consonants. A path from the cell $(0,0)$ to the cell $(N-1, M-1)$ is a sequence of the coordinates of the cells in the grid, through which one can move to reach $(N-1, M-1)$, starting from $(0,0)$. Two paths are said to be distinct if there is no cell common to both the paths except the starting cell and the destination cell. **Given a grid, Design a pseudocode to compute all the distinct paths from the top-left corner $(0,0)$ to the bottom-right corner $(N-1, M-1)$. Your pseudocode should be in two parts , (i) Pseudocode that computes all the appropriate paths, (ii) Pseudocode that computes all the distinct paths, by taking the output of the earlier pseudocode.** You may only move right, down, up, or left at each step. If a cell contains vowels , movement is not allowed through the cell. Your pseudocode should output the path as a sequence of the cells visited in the traversal. A path from $(0,0)$ to $(3,3)$ in the following grid is given by $\begin{bmatrix} f & b & d & i \\ e & a & l & o \\ s & t & t & u \\ d & t & p & b \end{bmatrix}$ $(0,0)-(0,1)-(0,2)-(1,2)-(2,2)-(3,2)-(3,3)$. Your pseudocode should output all the possible distinct paths. For the illustration of your pseudocode, you are supposed to take a grid with both N and M greater than 4 and the illustration should be in such a way that it yields a minimum of four paths and a minimum of two distinct paths. You are not supposed to use any illustration given in this question as part of your illustration in the answer. [**Rubrics:** Logic for pseudocode: 2 marks, Illustration for the pseudocode that computes all the appropriate paths: 4 marks, Illustration for the pseudocode that computes all the distinct paths : 2 marks, Pseudocode that computes all the appropriate paths : 3 marks, Pseudocode that computes all the distinct paths : 3 marks, Time-complexity :1 mark]	15	2	6
07.	Consider a positive integer d of length n with n-digits and a set S which consists of a few sub-integers of d. Some sub-integers of 234567 are 23, 34, 4567. **Design a greedy based recursive pseudocode and a greedy based iterative pseudocode to compute the maximum number of non-overlapping sub-integers from S such that no two selected sub-integers overlap in their positions in d.** If $d = 12345678$, $S = \{123, 234, 345, 456, 678, 34\}$, maximum number of sub-integers of d in Swhich does not over-lap is 2 and the those sub-integers are 123, 456. Your pseudocode should output the maximum number of sub-integers as well as those sub-integers. You have to illustrate your pseudocode by considering a set S with a minimum of eight elements. You are not supposed to use any illustration given in this question as part of your illustration in the answer. [**Rubrics:** Logic for pseudocode: 3 marks, Illustration for greedy-based recursive pseudocode : 4 marks, Greedy based recursive Pseudocode : 3 marks, Greedy based iterative pseudocode : 3 marks, Time-complexity of the greedy based recursive pseudocode :1 mark, Time-complexity of the greedy based iterative pseudocode :1 mark]	15	2	6

08. | 15 | 1,3 | 4

Algorithm 1 Find MFV in Flow Network

Require: Flow network $G = (V, E, c, s, t)$ with capacities c, source s, and target t
Ensure: Set of MFV in G

```
Initialize maxFlow ← 0
Initialize vertexVisitCount[v] ← 0 for each vertex v in V
Initialize MF ← 0
Initialize MFV ← ∅
while there exists an augmenting path P from s to t in G do
    Find the minimum residual capacity c_min along path P
    Update the flow along edges in P by c_min
    maxFlow ← maxFlow + c_min
    for each vertex v in path P (excluding s and t) do
        vertexVisitCount[v] ← vertexVisitCount[v] + 1
    end for
end while
for each vertex v in V do
    if vertexVisitCount[v] > MF then
        MF ← vertexVisitCount[v]
        MFV ← {v}
    else if vertexVisitCount[v] = MF then
        Add v to MFV
    end if
end for
return MFV
```

Understand the Algorithm 1 and answer the following.

a. Describe the functionality of the above algorithm. [3 marks]

b. Compute the time-complexity of the algorithm. [2 marks]

c. Consider a directed flow network $G = (V, E, s, c)$ with: Vertices: $V = \{s, v_1, v_2, v_3, v_4\}$, Edges with capacities c: $\{(s, v_1) : 10, (s, v_2) : 15, (v_1, v_3) : 5, (v_2, v_3) : 10, (v_2, v_4) : 10, (v_3, v_4) : 10\}$, source s. Given G , compute the output of Algorithm 1 with target vertex as v_4 [5 marks]

d. Will there be a change in the output if the target vertex is changed from v_4 to v_3 to the graph described in the question no (3)? If so, what will be the new output? [3 marks]

e. What is the purpose of the vertexVisitCount array in the algorithm, and how is it updated during the execution of the algorithm? [2 marks]

BL-Bloom's Taxonomy Levels - (1.Remembering, 2.Understanding, 3.Applying, 4.Analysing, 5.Evaluating, 6.Creating)

VIT®
Vellore Institute of Technology

Reg. No. : 22BRS1017

Final Assessment Test(FAT) - Nov/Dec 2024

Programme	B.Tech.	Semester	Fall Semester 2024-25
Course Code	BCSE204L	Faculty Name	Prof. Sivaramakrishnan N
Course Title	Design and Analysis of Algorithms	Slot	A2+TA2
		Class Nbr	CH2024250101368
Time	3 hours	Max. Marks	100

General Instructions

- Write only Register Number in the Question Paper where space is provided (right-side at the top) & do not write any other details.
- 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: Computation of T(n), Time-complexity.

Course Outcomes

1. Apply the mathematical tools to analyze and derive the running time of the algorithms.
2. Demonstrate the major algorithm design paradigms.
3. Explain major graph algorithms, string matching and geometric algorithms along with their analysis.
5. Explain the hardness of real-world problems with respect to algorithmic efficiency and learning to cope with it.

Section - I
Answer all Questions (4 × 10 Marks)

*M - Marks

Q.No	Question	*M	CO	BL
01.	Given a set of n points in $P : \{p_1, p_2, p_3 \dots p_n\}$ in a 2-dimensional plane, $f(p_i)$ be the value of each point p_i, $(1 \le i \le n)$ in P and is calculated as: $f(p_i) = \frac{(3x_i+4y_i)}{2}$. Given P, design a pseudocode using divide and conquer strategy to sort the n points based on the decreasing order of their values. For any two points, p_i and p_j in P, if $f(p_i) = f(p_j)$, then sort the points in the increasing order of x-axis. For example, if $p_i = (7,1)$ and $p_j = (3,4)$, then $f(p_i) = f(p_j) = 12.5$ and the sorted order is $(3,4)$, $(7,1)$. Illustrate your pseudocode for $n = 8$, $P = \{(5,2),(2,3),(3,4),(6,1),(4,3),(7,2),(1,6),(7,1)\}$. [Rubrics: Logic: 2 Marks, Illustration: 3 Marks, Pseudocode : 3 Marks, Time-complexity: 2 Marks]	10	2	3
02.	You are given a metal rod of length L with the price list P: $\{p_1, p_2, p_3, \dots, p_n\}$, where each p_i, $(1 \le i \le n)$ represents the price of a rod-piece of length i. Every time you cut the rod into pieces, a cutting cost c is incurred. Suppose you decide to cut the rod of length i into two pieces of length j and k, where $i = j + k$, then the profit P_i obtained by selling the piece of rod of length i is calculated as $P_i = p_j + p_k - c$. If we	10	2	4

cut the rod of length $L = 6$ into three pieces, say with length 3, 2, 1 (first we cut into pieces of length 3 each and the right piece is cut into length of 2 units, 1 unit respectively) then the profit obtained by selling the rod of length 6 is calculated as $P_6 = p_3 + p_2 + p_1 - 2c$, since the process involves two cuts. Like this, we calculate P_i based on the prices of the piece and the number of cuts involved in the process. Your task is to determine the maximum profit you can obtain by cutting the rod into smaller pieces and selling the pieces or selling it as a whole, accounting for these cutting costs. Given L, P, and c, design a pseudocode using dynamic programming to compute the maximum profit obtained by selling the rod of length L by cutting it into pieces. Illustrate your algorithm for $L = 6$ and $P = \{1, 5, 8, 9, 10, 17\}$, with $c = 2$.
[Rubrics: Logic: 2 Marks, Illustration: 3 Marks, Pseudocode : 3 Marks, Time-complexity: 2 Marks]

03. **Algorithm 1 ABC** 10 1 2

```
1: Input: n - size of the memory array
2: Output: total allocated, memory
3: function XYZ(n)
4:     memory ← [0] × n
5:     function PQR(i)
6:         if i ≥ n then
7:             return 0
8:         end if
9:         if i = 0 then
10:            memory[i] ← i + 1
11:            return PQR(i + 2) + memory[i]
12:        else
13:            return PQR(i + 1)
14:        end if
15:    end function
16:    total allocated ← PQR(0)
17: end function
18: return total allocated, memory
```

Understand the functionality of the above algorithm and answer the following:
a) Compute the memory array and the total allocated memory blocks for the input n= 10. [5 Marks]
b) Write the recurrence relation for the function PQR(i) given in the algorithm and compute the time complexity. [5 Marks]

04. Given n points in $P = \{p_1(x_1, y_1), p_2(x_2, y_2), p_3(x_3, y_3) \ldots p_n(x_n, y_n)\}$ in a 2-dimensional plane. A connection between any two points p_i,p_j is a line segment joining p_i - p_j. The cost to connect any two points $p_i(x_i, y_i)$, $(1 \leq i \leq n)$ and $p_j(x_j, y_j)$, $(1 \leq j \leq n)$ through a line segment is defined as: $cost(p_i, p_j) = |x_j - x_i| + |y_j - y_i|$. A connection-sequence of P is the sequence of line segments joining all the points P such that the end-point of any line segment is the start-point of the next line segment except the last line segment. There are more than one possible connection-sequences of P. For example, if $n = 4$, $P = \{p_1(0, 0), p_2(2, 3), p_3(5, 0), (p_4(10, 5)\}$, then $cost(p_1, p_2) = cost(p_1, p_3) = 5$, $cost(p_2, p_3) = 6$, $cost(p_2, p_4) = cost(p_3, p_4) = 10$, $cost(p_1, p_4) = 15$. The cost of one connection-sequence (p_1, p_2), (p_2, p_3),(p_3, p_4) is 21 and another possible connection-sequence (p_1, p_2), (p_2, p_4),(p_4, p_3) is 25. (p_1, p_2), (p_2, p_3),(p_3, p_4) is the connection-sequence with minimum cost 21. Given P, develop a suitable pseudocode to find the connection-sequence with minimum cost. Illustrate your pseudocode for $n = 6$, P= $\{p1(1, 1), p2(4, 2), p3(2, 5), p4(7, 3), p5(6, 6), p6(3, 8)\}$. 10 2 4

[Rubrics: Logic: 2 Marks, Illustration: 3 Marks, Pseudocode : 3 Marks, Time-complexity: 2 Marks]

Section - II
Answer all Questions (4 × 15 Marks)

*M - Marks

Q.No	Question	*M	CO	BL
05.	Given a set of n cities, $c = \{c_1, c_2, \ldots, c_n\}$, the distance between any two cities c_i, $(1 \le i \le n)$ and c_j, $(1 \le j \le n)$ are represented in the 2-dimensional matrix d, where $d[i,j]$ denotes the distance between c_i and c_j. The travel time between any two cities, c_i, and c_j, is represented in the 2-dimensional matrix t, where $t[i,j]$, denotes the travel time from c_i to c_j. A tour of the city c_i, is a journey that starts at city, c_i and traverses all the intermediate cities in c, say, $c_k, c_l, c_m, \ldots, c_j$ exactly once and reaches to c_i such that $\{i,k,l,m,\ldots,j \le n\}$. That is, $[c_i - c_k - c_l - c_m, \ldots, c_j - c_i]$ is the tour of the city c_i. The cost of the tour of the city c_i is calculated as $w_d \times [d(i,k) + d(k,l) + d(l,m), \ldots, d(j,i)] + w_t \times [t(i,k) + t(k,l) + t(l,m), \ldots, t(j,i)]$, where w_d and w_t, are the constant distance-weight and time-weight respectively. Given the values of c, d, t, w_d, and w_t, design an algorithm to compute the tour of city c_i such that both the cost and travel time of the tour are minimal. Illustrate your algorithm for $n = 4$, $w_d = 0.7$, $w_t = 0.3$, the starting city is c_1 with the following d and t values. $$d = \begin{bmatrix} 0 & 10 & 15 & 20 \\ 10 & 0 & 35 & 25 \\ 15 & 35 & 0 & 30 \\ 20 & 25 & 30 & 0 \end{bmatrix} \quad t = \begin{bmatrix} 0 & 5 & 50 & 50 \\ 5 & 0 & 10 & 50 \\ 50 & 10 & 0 & 8 \\ 50 & 50 & 8 & 0 \end{bmatrix}$$ [Rubrics: Logic: 3 Marks, Illustration: 5 Marks, Pseudocode : 5 Marks, Time-complexity: 2 Marks]	15	3	4
06.	Consider a flow network $F = (G, c, s, t)$, where $G = (V, E)$ is a directed graph with vertices V and directed edges E, s is the source vertex, and t is the target vertex. For each directed edge $(u, v) \in E$, the capacity function $c(u, v)$ is a non-negative integer defining the maximum flow from u to v. Given F design a suitable pseudocode (Pseudocode 1) to calculate the maximum possible flow from s to t. Illustrate your pseudocode for the following inputs: $V=\{s, v_1, v_2, v_3, v_4, t\}$, c : $\{s \to v_1 : 10, s \to v_2 : 8, v_1 \to v_3 : 5, v_1 \to v_4 : 5, v_2 \to v_3 : 7, v_2 \to v_4 : 4, v_3 \to t : 8, v_4 \to t : 6\}$ In the above given network F, certain edges in G exhibit a flow loss factor (f_{loss}), represents that a percentage of the flow sent across these edges is lost before it reaches the destination. For instance, consider the edge $v_1 \to v_2$ has a capacity of 8 units and f_{loss} of 20%. If we initially send 5 units along $v_1 \to v_2$, the effective flow reaching v_2 is calculated as $5 \times (100 - 20)\% = 4$, resulting in a loss of 1 unit. In a subsequent iteration, if we send 3 units along $v_1 \to v_2$, the effective flow reaching v_2 would be $3 \times (100 - 20)\% = 2.4$, with a loss of 0.6 units. This process is repeated until the flow on the edge $v_1 \to v_2$ exhausted. Modify the above algorithm to compute the maximum possible flow from s to t, by considering the flow loss factors (f_{loss}) on the specified edges in G. Given the instance of the flow network problem, consider there is a 20% flow loss on the edge $s \to v_2$. Analyze and justify the impact of flow loss on the maximum flow achieved in the network. [Rubrics: Logic : 2 Marks, Illustration for maximum flow calculation : 3 Marks, Pseudocode 1 (Maximum flow network) : 3 Marks, Illustration of pseudocode with flow loss rate : 3 Marks, Justification : 2 Marks, Time-complexity : 2 Marks]	15	3	4
07.	Let S is a string formed with elements from S_1, S_2, S_3 where $S_1 = \{a, b, \ldots, z\}$, $S_2 = \{A, B, \ldots, Z\}$ and $S_3 = \{0, 1, \ldots, 9\}$. S should consist of at least one element with more than one occurrence from each S_1, S_2, and S_3. Each element in S_1, S_2, and S_3 is assigned a value where values of $a,b,c,\ldots,z$ are $1,2,3,\ldots,26$, $A,B,C,\ldots,Z$ are $-1,-2,-3,\ldots,-26$, and $0,1,2,\ldots,9$ are $9,8,7,\ldots,0$ respectively. $val(S)$, the value of the string S is calculated as: $val(S) = \sum_{a \in S_1 \text{ or } a \in S_2 \text{ or } a \in S_3} val(a)$. That is, $val(S)$ is the sum of the values of the character that occurs in S, where the value of the characters	15	3	4

with multiple frequencies are counted only once. For example, if S = "CoComplex1515", then $val(S) = (-3) + 15 + 13 + 16 + 12 + 5 + 24 + 8 + 4 = 94$.
Given n strings, design a pseudocode to find out the following:
a) The strings S_{min} with minimum value and S_{max} with maximum value.
b) Modify S_{min} by appending the elements from S_1, S_2, S_3 such that $val(S_{min}) = val(S_{max})$
c) Modify S_{max} by appending the elements from S_1, S_2, S_3 such that $val(S_{max}) = val(S_{min})$
For example, if S_{min} = "XYXYxy55", S_{max} = "AbAb1010", then $val(S_{min}) = 4$ and $val(S_{max}) = 18$. Modification of S_{min} = "XYXYxy55g2" with value 18. Similarly modification of S_{max} = "AbAb1010CDG" with value 4.
[Rubrics: Logic : 2 Marks, Illustration: 3 Marks, Pseudocode 1: 4 Marks, Pseudocode 2: 4 Marks Time-complexity: 2 Marks]

08. a) Given n tasks, $t = \{t_1, t_2, \ldots, t_n\}$, each with an associated execution time $e = \{e_1, e_2, \ldots, e_n\}$. Divide the set t into two subsets so that both have the same execution time as half the total sum of all task execution time. If the total execution time is e_total, then the execution time of each subset must sum to $\frac{e\_total}{2}$. Given t and e, check if such a partition exists to find the subsets. Identify the complexity class (P/NP/NPC) of this problem with justification. [5 Marks]. 15 2,5 3

b) Consider a grid represented as a matrix G of size $p \times q$, where p and q are positive integers denoting the number of rows and columns, respectively. Each cell $G[i, j]$, contains an integer value that represents the points available in the cell. The player is awarded the points equal to the integer value in a cell, when the player enters the cell.
The constrains of the problem are as follows:

- A player begins at the top-left cell $[1, 1]$, and has to make exactly k moves to collect points in each of the k cells entered.
- During each move, the player can enter into an adjacent cell with left, right, up or down moves from the current cell.
- The player should not enter any cell which is already visited.
- The player has to start from cell $[1, 1]$ and make k moves in such a way that, the player makes exactly two successive down moves in addition to the other allowed moves.

Given the grid G and positive integer k, design a pseudocode using backtracking to find the path from the top-left cell with k moves (which includes, exactly two successive down moves) in such a way that the player is rewarded with maximum points. For example, consider the following grid of size 3×4 with $k = 4$. One possible path from $[1, 1]$ with 4 moves (including exactly two successive down moves) is $(1, 1) \rightarrow (1, 2) \rightarrow (2, 2) \rightarrow (3, 2) \rightarrow (3, 1)$, and the points awarded to the player is 20.

$$G = \begin{bmatrix} 2 & 3 & 1 & 4 \\ 0 & 6 & 2 & 1 \\ 4 & 5 & 2 & 8 \end{bmatrix}$$

[Rubrics: Logic : 2 Marks, Illustration: 3 Marks, Pseudocode: 3 Marks Time-complexity: 2 Marks]

BL-Bloom's Taxonomy Levels - (1.Remembering, 2.Understanding, 3.Applying, 4.Analysing, 5.Evaluating, 6.Creating)

⇔⇔⇔

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of the UGC Act, 1956)

Final Assessment Test(FAT) - Nov/Dec 2024

Programme	B.Tech.	Semester	Fall Semester 2024-25
Course Code	BCSE204L	Faculty Name	Prof. Revathi A R
Course Title	Design and Analysis of Algorithms	Slot	D1+TD1
		Class Nbr	CH2024250101208
Time	3 hours	Max. Marks	100

General Instructions

- Write only Register Number in the Question Paper where space is provided (right-side at the top) & do not write any other details.
- 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.

Course Outcomes

1. Apply the mathematical tools to analyze and derive the running time of the algorithms
2. Demonstrate the major algorithm design paradigms.
3. Explain major graph algorithms, string matching and geometric algorithms along with their analysis.
5. Explain the hardness of real-world problems with respect to algorithmic efficiency and learning to cope with it.

Section - I
Answer all Questions (4 × 10 Marks)

*M - Marks

Q.No	Question	*M	CO	BL
01.	Let $L =< B_1, B_2, \ldots B_n >$ be a list of n boolean expressions and let k be a positive integer with $k < n$. The operators in the boolean expressions are $AND(\wedge)$ and $XOR(\oplus)$ only. Write an algorithm that does not employ sorting but uses a Divide-and-Conquer technique to print all the boolean expressions $B_i \in L$ where the number of $XOR(\oplus)$ operations in B_i are greater than k. Illustrate the algorithm with your example and derive its time complexity. **[Rubrics: Logic- 2 marks, Algorithm-4 marks, Time complexity- 2 marks, and Example-2 marks]**	10	2	3
02.	Consider the following algorithm and answer the queries given below. **Count_Algo( )** ***K=0*** ***For I_1 = 1 to n***	10	1	4

For $I_2 = 1$ *to* I_1
For $I_3 = 1$ *to* I_2
....
....
For $I_m = 1$ *to* I_{m-1}
$K = K+1$

(a) Write the output of **Count_Algo( )** algorithm, when $n = 5$ and $m = 4$ **(3 marks)**
(b) Describe the functionality of the **Count_Algo( )** algorithm **(5 marks)**
(c) Compute the time complexity of the algorithm. **(2 marks)**

03. Let $P = \{p_1, p_2, \cdots p_n\}$ be a given set of n points in the two dimensional plane and let $CH(P)$ denote the convex hull of P. We define the convex layers of P recursively as follows: 10 3 3

1. $P_i = P, i = 0;$
2. $P_i = P_{i-1} - CH(P_{i-1}), i \geq 1,$ and $P_{i-1} \neq \emptyset.$

Here, we start with the initial set of given points $P_0 = P$, and obtain the subsequent convex layers of $P_i, i \geq 1$ (denoted by $CH(P_i)$) from P_{i-1} by removing the points belonging to $CH(P_{i-1})$. This process continues until the set P_i is empty for some i. Then, the convex layers are exactly the points $CH(P_{i-1})$ obtained for each value of i. If for some $k > 0$, P_k becomes empty then we have $k - 1$ convex layers of P. Given the points P as an input, design an algorithm to compute the convex layers of P.

[Rubrics: Logic- 2 marks, Algorithm-4 marks, Time complexity- 2 marks and, Example-2 marks]

04. Let $< P_1, P_2, \ldots, P_n >$ be a list of n points on a two-dimensional plane. Design algorithm to sort a sequence $<P_1, P_2, \ldots, P_n>$ of n points according to their polar angles. Your algorithm should take $O(nlogn)$ time and use cross-product to compare angles. 10 3 3

[Rubrics: Logic- 2 marks, Algorithm-4 marks, Time complexity- 2 marks, and Example-2 marks]

Section - II
Answer all Questions (4 × 15 Marks)

*M - Marks

Q.No	Question	*M	CO	BL

05. **The 2D_String –Matching Problem** is defined as follows. We assume that a text T is an $n \times n$ 2D array and pattern P is also an $m \times m$ 2D array, where $m \leq n$. We say that pattern P occurs in text T if ST_IS is a sub-matrixof T such that $ST\_I = P.S$ Given, a text T and a pattern P. the task is to find the total number of occurrences of P in T. For example, let 15 3 3

$$T=\begin{bmatrix}1 & 2 & 4\\ 1 & 2 & 6\\ 1 & 2 & 7\end{bmatrix} \text{ and } P=\begin{bmatrix}1 & 2\\ 1 & 2\end{bmatrix}.$$

The pattern P occurs in T, 2 times. Use the Rabin-Karp algorithm to design two algorithms that use different functions to solve the *2D_String_Matching* problem. Compare the two functions used to analyze and conclude which works better and why. Compute the time complexity of your algorithm and also Illustrate the algorithm with your example.

[Rubrics: Logic- (2+2) marks, Algorithm- (2+2) marks, Time complexity- (0.5 + 0.5) marks, Function Comparison and Analysis - (1+1) marks and Illustration - (2+2) marks]

06. Consider the following *3-PARTITION* problem. Given integers $<a_1, a_2, \cdots, a_n>$, we want to determine whether it is possible to partition $\{1, 2, \ldots, n\}$ into three disjoint subsets I, J, K such that $\sum_I a_i = \sum_J a_j = \sum_K a_k = \frac{1}{3}\sum_{i=1}^{n} a_i$. As an example, for the given input $< 1, 2, 3, 4, 4, 5, 8 >$, the answer is yes because there is a partition $< 1, 8 >$, $< 4, 5, >$, $< 2, 3, 4 >$. On the other hand, for input $< 2, 2, 3, 5 >$ the answer is 15 2 2

no. Design an algorithm using a dynamic programming approach to solve the *3-PARTITION* problem and compute the time complexity of your algorithm. Illustrate the algorithm with your example.

[Rubrics: Logic- 4 marks, Algorithm-7 marks, Time complexity- 2 marks, and Illustration-2 marks]

07. Let $G = (V, E)$ be a flow network with positive edge capacities. An edge in G is *upper-binding* if increasing its capacity by 1 also increases the value of the maximum flow by 1 in G. Similarly, an edge is *lower-binding* if decreasing its capacity by 1, also decreases the maximum flow value by 1 in G. 15 3 3

(a) Design an algorithm to find all *upper-binding* edges in G, given both G and a maximum flow in G as input.

(b) Design an algorithm to find all *lower-binding* edges in G, given both G and a maximum flow in G as input.

[Rubrics: Logics- (2+2) marks, Algorithms- (3+3) marks, Time complexity- (0.5 + 0.5) marks, and Illustration - (2+2) marks]

08. Consider the following **Puzzle-Peg** problem. The puzzle-peg problem is played on a board with $n \times n$ *board that contain* n^2 *slots*. Initially, all but one of the slots are occupied by pegs. That is, $n^2 - 1$ slots are occupied with pegs and one slot is empty. For example, a board with 16 slots is shown in the figure given below. The occupied slots are represented by dark solid circles and the empty slot is represented as a white hollow circle in the figure. Assuming that the slots are numbered from (1,1) up to (4,4), the empty slot is at (2,3). 15 2 3

A move is legal if,

a. the move is either horizontal or vertical and not diagonal,

b. a peg moves into an empty slot by jumping over exactly one peg that is adjacent to the empty slot. While doing so, the peg that was jumped over is removed from the board.

For e.g., in the figure given below the peg at slot (2,1) can jump over the peg at slot (2,2) and move into the empty slot. This move shifts the empty slot from (2,3) to (2,1) and the peg at slot (2,2) is removed from the board leaving. On the other hand, the peg at slot (4,3) can jump over the peg at slot (3,3) and move into the empty slot shifting the empty slot from (2,3) to (4,3), thereby removing the peg at slot (3,3). Note that in this example no other moves can be made other than the two moves given above. Also, for every move that is made, a new empty slot is created.

For each legal move that is made, a peg p_i is removed from the square if another peg p_j jumps over p_i to occupy an empty slot.

Design a backtracking algorithm for solving the following versions of this puzzle - Starting with a given location of the empty slot, find the shortest sequence of moves that eliminates 15 pegs with no limitations on the final position of the remaining peg.

[Rubrics: Logic-4 marks, Algorithm-5 marks, Example- 3 marks, Time Complexity - 3]

BL-Bloom's Taxonomy Levels - (1.Remembering, 2.Understanding, 3.Applying, 4.Analysing, 5.Evaluating, 6.Creating)

⇔⇔⇔

VIT®
Vellore Institute of Technology

Reg. No. : 22BCE1351

Final Assessment Test(FAT) - Nov/Dec 2024

Programme	B.Tech.	Semester	Fall Semester 2024-25
Course Code	BCSE204L	Faculty Name	Prof. Tapabrata Roy
Course Title	Design and Analysis of Algorithms	Slot	D2+TD2
		Class Nbr	CH2024250102302
Time	3 hours	Max. Marks	100

General Instructions

- Write only Register Number in the Question Paper where space is provided (right-side at the top) & do not write any other details.

Course Outcomes

1. Apply the mathematical tools to analyze and derive the running time of the algorithms
2. Demonstrate the major algorithm design paradigms.
3. Explain major graph algorithms, string matching and geometric algorithms along with their analysis.
5. Explain the hardness of real-world problems with respect to algorithmic efficiency and learning to cope with it.

Section - I

Answer all Questions (4 × 10 Marks) *M - Marks

Q.No	Question	*M	CO	BL
01.	Consider the following algorithm and answer the queries given below. Let $R = \{a_1, a_2, \ldots, a_r\}$ be a proper subset of $X = \{1, 2, 3, .. n\}$ with $a_1 < a_2 < \cdots < a_r$. INPUT: R and a positive value n. **COM_ALGORITHM**(R, n) 1. i=r 2. **While**(a_i==n-r+i) 3. i= i-1 4. a_i=a_i+1 5. **For** j= i+1 to r 6. a_j=a_i+j-i a. Write the output of **COM_ALGORITHM** if input n=6 and\ { a_1 = 1,a_2=2,a_3=5, a_4=6\} **(3 marks)** b. Describe the functionality of the **COM_ALGORITHM**(R,n) algorithm **(5 marks)** c. Compute the time complexity of the algorithm. **(2 marks)**	10	1	4

02.	The **Fibonacci sequence** is a sequence in which each number is the sum of the two preceding ones. That is, The **Fibonacci sequence is** 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ... Here we are assuming that the sequence starts at 1. **Fibonacci_Sorting** problem: Let A be an array of size n. Assume that the index values of array A start with 1. Then, the **Fibonacci index positions** are precisely those index values of A that are part of the Fibonacci Sequence, that is, index values 1, 2, 3, 5, 8, ... and so on. The **Fibonacci_Sorting** problem is defined as the process of organizing the elements of A, that occur at the Fibonacci index positions (1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...), in decreasing order. The elements occurring at other index positions are not disturbed. Design an algorithm using the *divide-and-conquer* technique to solve the **Fibonacci _Sorting** problem. Illustrate your algorithm for any sample input and compute the running time of your algorithm. **[Logic – 2 marks, Algorithm – 3 marks, Illustration – 3 marks, Time Complexity – 2 marks]**	10	2	3
03.	Two non-parallel line segments in a 2D plane are called *skewline* segments if they do not intersect. Given n line segments in a 2D plane, design an algorithm to decide a pair of skew line segments in the given n line segments. Assume that the coordinates of the endpoints of each line segment are given. **[Logic – 2 marks, Algorithm – 3 marks, Illustration – 3 marks, Time Complexity – 2 marks]**	10	3	3

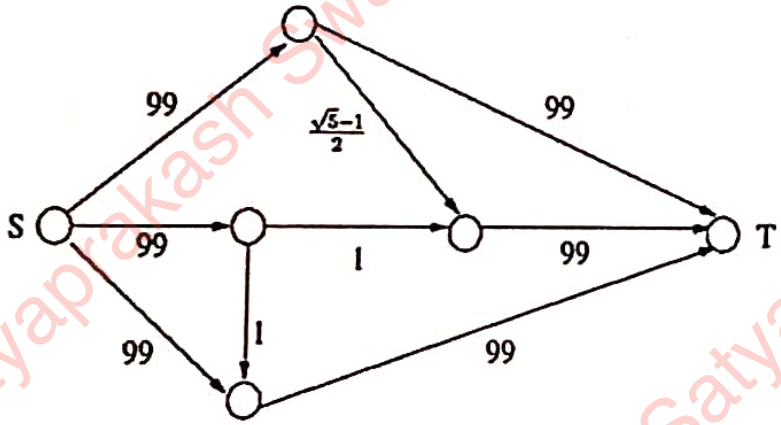
04.	Consider the "Disjoint Sets Problem" described as follows: Given two sets S_1 and S_2, the problem is to check whether the sets have a common element or not. a. Compute the complexity class (P class or NP class or NP-complete class) of the "Disjoint Sets Problem". [5 marks] b. Consider the problem ***k-element Subset*** described as follows: Let $N = \{1, 2, \ldots, n\}$ be a set. Given N and k such that $1 \leq k \leq n$, the **k-element Subset** problem is to generate all subsets of N each containing at most k elements. Compute the complexity class(P / NP / NP-C) of the problem and justify your answer. [5 marks] **[Identification of class complexity - (2+2) marks, Justification with example - (3+3) marks]**	10	5	4

Section - II

Answer all Questions (4 × 15 Marks) *M - Marks

Q.No	Question	*M	CO	BL
05.	Consider the *MINIMUM UNIT SQUARE COVER* problem described as follows: Let P be a set of n points in the 2D plane. We say that a point $p \in P$ is covered by a square s, if p is either on the boundary or inside the square s. Every square is represented by four tuples - $(x_1, y_1)(x_1, y_2), (x_2, y_1)$ and (x_2, y_2) in the 2D plane. A square cover of P is a set S of unit squares (squares of size 1×1) such that any point $p \in P$ is covered by at least one square $s \in S$. The problem is to find a square cover of P with a minimum number of unit squares. Compute the complexity class of the problem *MINIMUM SQUARE COVER*. According to the complexity class you have arrived at, design an algorithm for the problem. Your algorithm should not be an exponential time. **[Logic – 2 marks, Algorithm – 3 marks, Illustration – 3 marks, Time Complexity – 2 marks, Complexity Class with Justification - 5]**	15	5	4

06.	Let $T_1, T_2, \ldots, T_k$ and $P_1, P_2, \ldots, P_k$ be k texts and patterns that are given. Assume that $\|T_i\| = n_i$ and $\|P_i\| = m_i, 1 \leq i \leq k$. Let $(T_i, P_j), 1 \leq i, j \leq k$ represent a tuple that defines a combination of a text T_i and a pattern P_j. Define the **Tuple-Pattern-Matching** problem as the problem of checking if pattern P_j occurs in text T_i for all values of i and j with $1 \leq i, j \leq k$. Further, we define the **Spurious-Tuple-Pattern-Matching** problem as the problem of finding a tuple with maximum spurious hits and a tuple with minimum spurious hits among all possible tuples. Modify the Rabin-Karp algorithm to output the total number of maximum spurious hits and minimum spurious hits among all tuple combinations. Note that in the case where $k = 2$, the possible tuple combinations are $(T_1, P_1), (T_1, P_2), (T_2, P_1)$ and (T_2, P_2). **[Logic – 3 marks, Algorithm – 6 marks, Illustration – 4 marks, Time Complexity – 2 marks]**	15	3	4
07.	**Max-Sum-Increasing-Subsequence Problem:** Let S be a sequence of n integers. A subsequence S' of S is called as a $Max - Sum - Increasing - Subsequence$, if the sum of the elements of S' is as large as possible and the elements of S' are arranged in non-decreasing order of their value. Such a subsequence is not necessarily contiguous (i.e., subsequences are not required to occupy consecutive positions within the original sequence of integers) or unique. For example, consider the sequence S = {0, 8, 4, 12, 2, 10, 6, 14, 1, 9, 5, 13, 3, 11}. The $Max - Sum - Increasing - Subsequence$ is {8, 12, 14} having sum 34, which is also the largest sum among all other $Max - Sum - Increasing - Subsequence's$ of S. Design an efficient algorithm to output a $Max - Sum - Increasing - Subsequence$ of a given sequence S. **[Logic – 3 marks, Algorithm – 6 marks, Illustration – 4 marks, Time Complexity – 2 marks]**	15	2	3

08.	Let $G = (V, E)$ be a flow network. Assume s is the source node and t is the sink node in G. An augmenting path P, is a path from s to t where every edge in P has positive capacity. The length of P, denoted as $l(P)$ is the number of edges in P. a. Design an algorithm to output all augmenting paths in G that have the same length. Compute the running time of your algorithm. **[10 marks]** b. Consider the network in the figure below where the source and the sink are marked as s and t, and the capacity of every edge is indicated. Will the Ford-Fulkerson method terminate on this network? Justify your answer. **[5 marks]**  **[a. (Logic– 3 marks, Algorithm– 4 marks, Illustration– 2 marks, Time complexity– 1 marks), b. Reasoning – 5 marks]**	15	3	3

BL-Bloom's Taxonomy Levels - (1.Remembering, 2.Understanding, 3.Applying, 4.Analysing, 5.Evaluating, 6.Creating)