VIT

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

Reg. No.:

Name :

Continuous Assessment Test II – Mar 2024

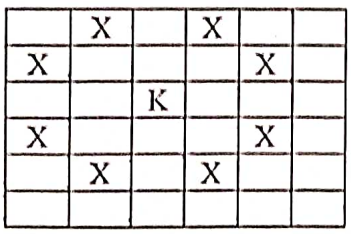
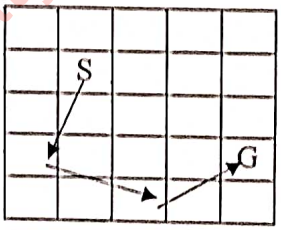
Programme	:	B. Tech	Semester	:	WINTER 2023-24
Course Code	:	BCSE204L	Class Nbr	:	CH2023240503369 CH2023240502397 CH2023240502398 CH2023240502399
Course Title	:	Design and Analysis of Algorithms			
Faculty	:	Kavipriya G, Jayaram B, Pavithra L K, Jannath Nisha O S.	Slot	:	A2 + TA2
Time	:	90 Minutes	Max. Marks	:	50

Answer all the Questions

Instructions:

- **Derive / Justify the time complexity of every algorithm written in the answer.**
- **Each answer should contain the following components: Logic, Illustration, Pseudocode, Time complexity.**
- **Any assumptions made are to be specified / defined clearly.**
-

Q. No.	Questions	Marks
1.	Let $U = \{x_1, x_2, x_3, \dots, x_n\}$ be a universe of elements and let S be a finite collection of subsets of U. A set $S_i \in S$ is said to cover an element x_j in U if x_j is in $S_i, \forall i, j$. An ordered sequence $X = \{S_1, S_2, S_3, \dots, S_r\}$, is called as a covering sequence of U of length r, if for each S_i in X, S_i covers at least one element of U that is not covered by $S_1 \cup S_2 \cup S_3 \cup \dots \cup S_{i-1}$, that is $\left\lvert S_i - \bigcup_{k=1}^{i-1} S_k \right\rvert \geq 1$ a. Given U and S, write an algorithm using the greedy technique that will find an X, such that X is the longest covering sequence of U. b. Give an example to show that the greedy approach will not find the optimum answer. As an example, if $U = \{1,2,3, \dots, 10\}$ and $S_1 = \{1,2,3\}, S_2 = \{2,4,7,10\}, S_3 = \{1,5,6,7\}, S_4 = \{1,3,9,10\}, S_5 = \{2,4,7,8\}, S_6 = \{1,5,10\}, S_7 = \{2,6,9\}$, then $X = \{S_1, S_7, S_6, S_5\}$ is a covering sequence of length 4 but not the longest sequence. $X = \{S_1, S_7, S_6, S_2, S_5\}$ is a covering sequence of length 5 which is a longest covering sequence. It is to be noted that there may be more than one covering sequence of the same length and also more than one longest covering sequence. The aim is to find one longest covering sequence. Your algorithm should contain all the design components such as, pseudo code, time complexity, logic and an illustration. **DO NOT write any code whatsoever.**	**15**

2.	Let $A[1 \ldots n]$ be an array of positive integers not necessarily in sorted order as a whole. Use the dynamic programming technique to write a recursive algorithm that will count the number of sub-arrays of A that are in non-decreasing sorted order. Two integers a_i, a_j, with $i < j$, are said to be in non-decreasing order if $a_i \leq a_j$. For example, if A = [6,5,1,2,4,3,4,5,6,4,7,3], then [1,2,3,5,6,7] is one of the non-decreasing subarrays. However this is not the largest subarray. [1,2,4,4,5,6,7] is the largest non-decreasing subarray. Your algorithm should contain all the design components such as, pseudo code, time complexity, logic and an illustration. **DO NOT write any code whatsoever.**	**10**
3.	Let A be an $n \times n$ grid with $n \geq 3$. For any knight located at cell c(i,j), the accessible cells of the knight are defined as follows: $$c(i,j) \xrightarrow{moves} \begin{cases} c(i \pm 2, j \pm 1) \\ c(i \pm 1, j \pm 2) \end{cases} \rightarrow (1)$$ Each of the eight moves defined in equation (1) is called **valid** if the resulting cell is within the $n \times n$ grid, and called **invalid** otherwise. Let S be a starting cell located at (i, j) and let G be the goal cell located at (p, q). S and G are given as inputs along with the size of the grid n. Write a recursive algorithm using the backtracking technique that will check if a knight starting at S can reach G following a sequence of finite number of valid knight moves. If there is a path from S to G then your algorithm should output "yes". If the knight cannot reach G from S, then the algorithm should output "No". As an illustration to the knights move defined in (1), consider the (6x6) grid in figure (a) where a knight K is placed in the cell (3,3). Then the knight K can move to any one of the cells marked X. As an illustration to the question, consider the (5x5) grid given in figure (b). From S, it is clear that the knight can reach G. Your algorithm should contain all the design components such as, pseudo code, time complexity, logic and an illustration. **DO NOT write any code whatsoever.**  (a)  (b)	**15**
4.	Let T be any given text of length n. We are usually required to check if a given pattern P occurs in T or not. If yes, we output the shifts at which this pattern P occurs. Instead, given only the text T, design an algorithm to find the longest pattern P that occurs maximum number of times in T. Know that $1 \leq \lvert P \rvert \leq n$. For example consider the text $T = abcdabc$. Here the longest pattern that repeats the maximum number of times is "abc". Your algorithm should contain all the design components such as, pseudo code, time complexity, logic and an illustration. **DO NOT write any code whatsoever.**	**10**

Reg. No.:

Name :

VIT

Vellore Institute of Technology

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

Continuous Assessment Test II– July '23

Programme	:	B.Tech.(CSE)	Semester	:	Fall Inter
Course Title	:	Design and Analysis of Algorithms	Code	:	BCSE204L
			Slot	:	C2 + TC 2
Faculty (s)	:	Dr. Manimegalai T Dr. Kalaipriyan T Dr. Om Kumar C.U. Dr. A.R Revathi Dr. Muthukumaran K Dr. Pavithra L K	Class Nbr(s)	:	CH2022232501098, CH2022232501099, CH2022232500927, CH2022232500935, CH2022232500937, CH2022232500928
Time	:	90 Minutes	Max. Marks	:	50 marks

Instructions:

- ✓ Answer all the FIVE questions.
- ✓ If any assumptions are required, assume the same and mention those assumptions in the answer script.
- ✓ 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.

---

1. You are given an encoded message consisting of digits from '0' to '9'. Each digit can be mapped to a corresponding letter from 'A-1', 'B-2', ... and 'Z-26' using a specific mapping. To decode the message, the digits must be grouped in a valid way and mapped back into letters based on the reverse mapping. For example, consider the string "11106". It can be decoded in multiple ways, such as "AAJF" with the grouping (1 1 10 6) or "KJF" with the grouping (11 10 6). However, the grouping (1 11 06) is invalid because the digit "06" cannot be mapped into a letter since it is different from "6". Design an algorithm that returns the total number of valid ways to decode a particular string and analyse its time complexity. **10**

DP - decode ways Q

Rubrics

Logic (2), Pseudocode(3), Illustration (3), Time Complexity (2)

2. Given the dimension of 'n' matrices, write an algorithm to form a compatible sub chain of matrices to perform matrix multiplication. There may be more than one compatible sub chain for the given chain of matrices. You may pick any one sub chain of length more than 3. Sub chain may include 'm' matrices in the order as given in the input, where m<=n. Using dynamic programming, parenthesize the sub chain obtained in such a way that minimum number of scalar multiplications are required. For example, if there are eight matrices with following dimension: **10**

MCM

A1 - 30 x 35 A2 - 30 x 25 A3 - 35 x 15 A4 - 10 x 20 A5-15 x 5 A6 - 5 x 10 A7 - 10 x 20 A8 - 20 x 25

Then compatible sub chain of matrices is A1, A3, A5, A6, A7, A8 and the minimum number of scalar multiplications required is 15125.

Rubrics

Logic (2), Pseudocode (3), Illustration (3), Time Complexity (2)

3. In a square grid of size N * N, a rat is initially positioned at cell (0, 0) and needs to reach the destination at cell (N - 1, N - 1). The rat can only move in four directions: up, down, left, and right. Some cells in the grid are blocked (denoted by 0) and the rat cannot move through them, while other cells are open (denoted by 1) and the rat can traverse through them. However, there is a twist - there are two special cells marked as "teleporters" (denoted by T) scattered throughout the grid. When the rat encounters a teleporter cell, it may or may not jump to the other teleporters cell on the grid.

1	0	1	T
1	1	1	0
0	1	T	1
1	1	1	1

10

n queen and backtracking

Devise an algorithm to find all possible paths that the rat can take to reach the destination.

Rubrics

Logic (2), Pseudocode(3), Illustration (3), Time Complexity (2)

4. 'Pick letters' is a kid's game in which a one-dimensional board of size 'n' and a box of 's' letter pieces. There are 'm' different letters in the box. Each letter in the box has a dimension and a score. The kid has to select few letters from the box in such a way that:

(i) Total dimension of the letters chosen do not exceed 'n'

knapsack - unbounded

(ii) Sum of the scores of the letters chosen is maximum

(iii) Number of vowels do not exceed 20% of n

(iv) He may pick one or more of the same letter but should not exceed the number of letter pieces of the selected letter

For example, if the dimension of the board is 50, and there are five letters with dimension and score as given below:

10

Letter	Number of Letter Pieces	Dimension	Score
A	5	5	8
B	3	4	12
E	8	2	10
D	4	3	4
I	10	2	3

Then the letters chosen will be B-3, E-8, A -2, D - 4 with maximum score as 148.

Rubrics

Logic (2), Pseudocode(3), Illustration (3), Time Complexity (2)

5. Consider the following algorithm.

```
Algo XXX (T, P)
    N=T.length
    M=P.length
    for i=0 to N-M
        count=0
        for j=1 to M
            if T[i+j] ! = P[j]
                count++
            if count<=2
                print ("exists at i")
    return count,i
```

string matching

10

i. Trace and write the output for the following input
   a. T = ACAACAAZBXCYAAB; P = ABAB [2 Marks]
   b. T = XYZJXYZXYZJ; P = IYXJ [2 Marks]

ii. Compute the time complexity of the Algo XXX ( ) [2 Marks]

iii. Describe the functionality of the above algorithm with the description for each line of the algorithm [2 Marks]

iv. Prepare an algorithm YYY( ) that does the same job as of Algo XXX( ) and prove that YYY( ) algorithm is efficient than algorithm XXX( ) [2 Marks]

VIT

Vellore Institute of Technology

Continuous Assessment Test II - March 2023

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

Instructions:

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

---

1. Given the sequence X : $ABBCDDGFFD$, Y : $ABBCDDFF$, the **Longest Common Pair Subsequences(LCPS)** of X and Y is $BBDDFF$. As an illustration, the characters of $BBDDFF$ occurs in both X and Y in such a way that BB, DD, FF occur in pairs. So in any LCPS of X and Y, the first two characters occur in both X and Y as adjacent to each other, the second pair of characters occur in both X and Y as adjacent to each other and so on. Given two sequences $< X_m, Y_n >$ (X_m is a sequence of m characters, Y_n is a sequence of n characters) m, n are both even integers, design a dynamic programming based pseudocode to compute the Longest Common Pair Subsequence (LCPS) of X_m and Y_n. Your design component should contain all the required steps. Analyse the algorithm with all the required steps. [10 Marks]

2. Given a square board B of size $n \times n$, design a back-tracking based pseudocode to place all the given n integers: $1, 2, 3, ...n$ in B such that (i) no successive numbers are placed in both the leading diagonals (in leading diagonals: line drawn from the left-most top cell to the right-most bottom cell as well as the line drawn from the left-most bottom cell to the right-most top cell) (ii) non-consecutive numbers can be placed in the same row or column of the board B. Your design component should contain all the required steps. Analyse the algorithm with all the required steps. [10 Marks]

3. Given a set of courses $S = \{C_1, C_2, ..., C_n\}$, where each C_i is represented by a pair of numbers , $C_i =< start - No., End - No. >$. The courses in S are the courses in which students are interested to take if offered by the school. $C_4 =< 1234, 1267 >$ means that the students with roll numbers 1234, 1235,

1236,...1266, 1267 are interested in taking the course. Assume that there are enough slots to offer all the n courses. Given S and a positive integer k, design a greedy based algorithm to identify the courses that can be offered to the students in such a way that (i) no course can have more than k students. (ii) one student can take only one course (iii) Maximum number of students are offered with the courses. Your design component should contain all the required steps. Analyse the algorithm with all the required steps. [10 Marks]

4. Consider the following algorithm.

Algorithm 1 XXXX

```
0: Input : X, Y
1: n=X.length
2: m= Y.length
3: MCh=" "
4: Mc=0
5: for i= 1 to m  do
6:    c=0
7:    for j= 1 to n  do
8:       if X[j]==Y[i] then
9:          c++
10:      end if
11:      if c > Mc then
12:         Mc=c
13:         MCh=Y[i]
14:      end if
15:   end for
16: end for
17: Return Mc, MCh
```

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

(a) Write the output of the algorithm when X=123235, Y = 137 [2]

(b) Write the output of the algorithm, when X= ABCDDCEADBDG, Y= ADDG, [2]

(c) Describe the functionality of the above algorithm with the description for each line of the algorithm . [4]

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

5. Propose a problem in detail (of your choice)which is not discussed in the classroom or in the lab sessions. Design a pseudocode A to solve the problem proposed by you that uses a non-recursive brute-force strategy. Note that A should not involve recursive functions. Transform your pseudocode A into an equivalent pseudocode B that uses dynamic programming strategy. Compute the time-complexity of both the pseudocodes A and B and identify the efficient one. Note that you are required to propose a problem for which a dynamic programming based pseudocode is possible.
[10 Marks]

$\Longleftrightarrow \Longleftrightarrow \Longleftrightarrow$

Registration Number : 22BAI1365

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

Continuous Assessment Test 2 - October 2024

Programme	B.Tech.(CSE)	Semester	Fall 2024-25
Course	Design and Analysis of Algorithms	Code	BCSE204L
Faculty	Dr B Indira Dr G Kavipriya Dr N Sivaramakrishnan Dr D Selvam Dr J Omana Dr M Raja	Slot/Class No.	A2+TA2 CH2024250100959 CH2024250100963 CH2024250101368 CH2024250101371 CH2024250101375 CH2024250102307
Time	90 Minutes	Max. Marks	50

Instructions:

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

1. Consider a positive integer n representing the size of an $n \times n$ matrix denoted as L, where each entry in L_{ij} belongs to the set $\{1, 2, \ldots, n\}$. You are given a partially filled matrix L as follows:

$$L = \begin{bmatrix} 1 & - & - & - \\ - & 2 & - & - \\ - & - & 3 & - \\ - & - & - & 4 \end{bmatrix}$$

Your task is to complete the partially filled matrix by ensuring the following constraints: Each number appears exactly once in every row. Formally, for any row $i \in \{1, 2, \ldots, n\}$, the set of entries in that row, $\{L_{i1}, L_{i2}, \ldots, L_{in}\}$, must contain no repeated elements and must be equal to $\{1, 2, \ldots, n\}$. Each number appears exactly once in every column, for any column $j \in \{1, 2, \ldots, n\}$, the set of entries in that column, $\{L_{1j}, L_{2j}, \ldots, L_{nj}\}$, must contain no repeated elements and must also be equal to $\{1, 2, \ldots, n\}$. Apply a suitable algorithm to explore all possible ways to fill the incomplete matrix by satisfying the above constraints. [10 marks]

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

2. Let x and y be two odd integers. A string $S[1 \ldots x]$ is said to be a First-Half Match if the first half of S appears somewhere in the second half of a text $U[1 \ldots y]$. Similarly, S is said to be a Second-Half Match, if the second half of S appears in the first half of U. For example, if S = ABCDE, then the first half of S is AB, and the second half is CDE. If U = XYCDEFGAB, AB is found in the second half of

U at position 7 (First-Half Match), and **CDE** is found in the first half of U at position 2 (Second-Half Match). Design a pseudocode that determines whether S is a First-Half Match, a Second-Half Match, or both, and calculate the corresponding shift (starting index) of the match in U. If no match is found, the output should be none. [10 marks]

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

3. Given a set of symbols S = {A, B, C, D, E, F, G, H} and their corresponding frequencies F = {7, 3, 8, 2, 6, 5, 12, 10}. You are tasked with creating an efficient Huffman encoding scheme. However, instead of a traditional binary tree, you are asked to design a tree with a variable branching factor b (where $b \geq 3$). Compute the total cost of encoding and compare the efficiency of the proposed algorithm with traditional binary approach for the same set of symbols. [10 marks]

4. Consider the given algorithmS

Algorithm 1 F1(S_1, S_2)

1: **Input:** S_1, S_2
   {S_1, S_2 are strings of the same length, and _ indicates an empty string}
2: $n = S_1$.length()
3: $d\_table[1,..,n] =$ _
4: $d\_table[1] = S_1$
5: $d\_table[2] = S_2$
6: **return** $(F2(n, d\_table)$

Algorithm 2 F2(n, d_table)

1: **if** $d\_table[n] \neq$ _ **then**
2: **return** $d\_table[n]$
3: **end if**
4: S_x = F3(F2($n-2, d\_table$), F2($n-1, d\_table$))
5: $d\_table[n] = S_x$
6: **return** S_x

Algorithm 3 F3(S_1, S_2)

1: $S_x =$ _
2: $m = S_1$.length()
3: **for** $i = 1$ to m **do**
4: **if** $i\%2 \neq 0$ **then**
5: $S_x = S_x + S_1[i]$
6: **else**
7: $S_x = S_x + S_2[i]$
8: **end if**
9: **end for**
10: **return** S_x

(a) For the input ABCD, DEFG, compute the output of the algorithm. [3 marks]

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

(c) The above algorithm is modified by deleting line 5 in **Algorithm 2**. Compare the complexity of the modified algorithm with the original algorithm. [4 marks]

5. There are a row of n coins of values $\{v_1, v_2, v_3, \ldots, v_n\}$; the objective is to pick up the maximum amount of money subject to the constraint that no two coins adjacent in the above list can be picked up. Develop a dynamic programming solution for this optimization problem. For e.g. if the values of $n = 6$ coins given are $\{5, 1, 2, 10, 6, 2\}$, then the coins that need to be picked up to obtain the maximum sum of 17 are $C1 = 5, C4 = 10, C6 = 2$ [10 marks]

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

$\Longleftrightarrow \Longleftrightarrow \Longleftrightarrow \Longleftrightarrow$

Registration Number : 22BPS1059

VIT
Vellore Institute of Technology
CHENNAI

Continuous Assessment Test II - October 2024

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

Instructions:

- Answer all the FIVE questions.
- If any assumptions are required, assume the same and mention those assumptions in the answer script.
- Use of intelligence is highly appreciated.
- In this open-book exam, you are expected have a maximum-thinking and answer the question with just the necessary required information.
- You are requested to answer the five questions in the first five pages of your answer book, with one question each in a page.
- You are requested to do all the rough works from the page six. Contents from page six onwards, will not be evaluated.

---

1. You are given two sequences which $X =< A_1A_2, ...A_m >$, $Y =< B_1B_2, ...B_n >$ where A_i's and B_j's are matrices, $0 < i \leq m$, $0 < j \leq n$. The order of the matrix A_i, is represented as a 2-tuple (rA_i, cA_i), where rA_i is the number of rows in A_i and cA_i represents the number of columns of A_i. Similarly, the order of the matrices B_i, $i = 0$ to n is represented as a 2-tuple (rB_i, cB_i). The task is to compute the Longest-Common-Subsequence of X and Y, in the sense of the order of the matrices, denoted by $LCS(X, Y)$. For example, if $X =< A_1A_2A_3A_4, A_5 >$, with sizes $(5, 10), (7, 3), (3, 12), (12, 5), (5, 7)$ respectively, $Y =< B_1B_2B_3B_4 >$ with sizes $(7, 3), (3, 12), (5, 7), (4, 7)$, LCS(X,Y) is $A_2A_3A_5$. Here $A_2 \cong B_1$, $A_3 \cong B_2$, $A_5 \cong B_3$. Here $A_i \cong B_j$, for any i, j, means that the matrix A_i and B_j are of the same order. We can also write $LCS(X, Y)$ as $B_1B_2B_3$. For the purpose of this problem, $LCS(X, Y)$ should be written in terms of A's alone. You are required to use the dynamic programming strategy for this problem. The sequence $< X, Y >$ is represented as $[m, n]$, where the sequence X has m matrices and Y has n matrices. Length of the $LCS(X, Y)$ is represeneted as $L[m, n]$,

   (a) Write the recurrence relation used to compute the $L[m, n]$ [2 marks]

   (b) What will be the entry in the $[3, 2]^{th}$ cell of the bottom-up memoization table used for the dynamic programming strategy in the computation of $L[m, n]$, if the order of matrices of X are : $(2, 3), (7, 9), (9, 6), (6, 8), (8, 18)$ and the order of the matrices of Y are: $(9, 6), (6, 8)(7, 8), (8, 18)$. [2 marks]

   (c) Mention the direction (left or top or diagonal) in the $(2, 3)^{th}$ cell for the input given in question 1(b). [2 marks]

Algorithm 1 LCS(X,Y)

d)

```
m=X.length
n=Y.length
Let b[0,...m:0,...,n] and c[0,...m:0,...,n] be two new tables, intialised with zero and 'space respectively
.....
.....
for i = 0 → m do
   for j = 0 → n do
      if ____(1)____ then
         c[i, j] = ____(2)____
         b[i, j] = 'diagonal'
      else if ____(3)____ then
         c[i, j] = ____(4)____
         b[i, j] = 'top'
      else
         c[i, j] = ____(5)____
         b[i, j] = 'left'
      end if
   end for
end for
```

There are four 'fill-in the blanks' in the Algorithm 1, which solves the given problem. Please write the response for 1.d.1 (1 mark), 1.d.2 (1 mark), 1.d.3 (1 mark), 1.d.4 (0.5 mark), 1.d.5 (0.5 mark) respectively, in separate lines. [4 marks]

2. Consider the 'Maximum Event Participation ' (MEP problem) described as follows : You are organizing a series of community events : $E_1, E_2, ..., E_n$, n is a positive integer. Each event runs for specified duration. Let the $d_1, d_2, ..., d_n$ be the duration (in minutes) of $E_1, E_2, ..., E_n$ respectively. A Participant can only attend events in the order they are listed and a participant can attend the events for a maximum duration D minutes. Given, $E_1, E_2, ..., E_n$, $d_1, d_2, ..., d_n$ and D, Task is to determine the maximum number of events that can be attended by a participant without exceeding D. Algorithm 2 is designed to solve the MEP problem.

Algorithm 2 MEP(E,d,D)

a)

```
1: E = [E_1, E_2, ..., E_n]
2: d = [d_1, d_2, ..., d_n]
3: SE = {} // {Set of Selected events}
4: CT = 0 // {Total duration of the selected events}
5: for ____(1)____ do
6:    ____(2)____
7:    ____(3)____
8:    ____(4)____
9: end for
10: return SE
```

There are four 'fill-in the blanks' in the Algorithm 2. Please write the response for 2.a.1, 2.a.2, 2.a.3, 2.a.4 respectively, in separate lines. [4 marks]

(b) What will be returned by the MEP pseudocde if d=[5,7,4,3,6,1] and D=17. [2 marks]

(c) Given D=15, compute n and $d = (d_1, d_2, ..., d_n)$ such that the following conditions are satisfied. [2 Marks]

- n is maximum (in the sense that d cannot be of size greater than n that satisfies the other required conditions)
- d is in decreasing order of d_i's and all d_i's are distinct integers.

(d) Describe the time-complexity of the above MEP pseudocode. [2 marks]

3. Given a text T and a pattern P, Compute the length of the Longest Common Subsequence of T and P such that LCS(T, P) occurs in T from the leftmost character to the rightmost and the LCS(T,P) Occurs in P from the rightmost to the leftmost. You are provided with two algorithms : Algorithm 3, Algorithm 4 to compute LCS(T,P).

(a)

Algorithm 3 $LCS(T, P)$

```
1: m = length(T)
2: n = length(P)
3: PR = _____(1)_____
4: Let L[0,...m:0,...,n] initialized to 0
5: for i= 1 to m do
6:    for j=1 to n do
7:       if _____(2)_____ then
8:          _____(3)_____
9:       else
10:         _____(4)_____
11:      end if
12:   end for
13: end for
14: return L[m][n]
```

There are four 'fill-in the blanks' in the Algorithm 3. Please write the response for 3.a.1, 3.a.2, 3.a.3, 3.a.4 respectively, in separate lines. [4 marks]

(b)

Algorithm 4 BruteForce-LCS

```
1: Input: Text T, Pattern P
2: Output: Length of Longest Common Subsequence
3: m ← length(T)
4: n ← length(P)
5: RP ← _____(1)_____
6: max_LCS_length ← 0
7: AST ← GenerateSubsequences(T)
8: ASP ← _____(2)_____
9: for _____(3)_____ do
10:   for _____(4)_____ do
11:      if A = B then
12:         _____(5)_____
13:      end if
14:   end for
15: end for
16: return max_LCS_length
```

Assusme that the Algorithm 4, is supported by a function GenerateSubsequences(). If you want any other functions, you can assume the same. There are six 'fill-in the blanks' in the Algorithm 4. Please write the response for 3.b.1(1 mark), 3.b.2(1 mark) 3.b.3 (1 mark), 3.b.4 (1 mark), 3.b.5(2 marks) respectively in separate lines. [6 marks]

4. Given $S = \{a_1, a_2, ..., a_n\}$ where a_i's are activities with the respective start-time and finish-time as (s_i, f_i), and a pattern $P = \{(s_a, f_a)\}$, Task is to compute the maximum subset S' of S which has mutually compatible activities such that S' has at the least one activity which has the time interval described in $P = (s_a, f_a)$ within itself. For example, the activity $(2, 7)$ has the time interval $(3, 5)$, within itself. If $S = \{(2,7), (3,5), (6,8), (8,12), (7,17)\}$, $P = \{(3,5)\}$, then $S' = \{(3,5), (6,8), (8,12)\}$. One

approach to solve this problem is as follows: From S and P, we first compute a set F which will have the activities that has the time interval of P, with in itself. If F is non-empty, that means that S has actvities which have the time interval of P with in itself. In the above example, $F = \{(3,5),(2,7)\}$. In that case, we can find the maximum compatible subset of S, which will ensure that S' has atleast one activity which will have the time interval of P with in itself. Based on this method, Algorithm 5 is designed to compute S'.

Algorithm 5 Maximum Compatible Subset with Overlapping Activity

a)
1: **Input:** Set of activities $S = \{(s_i, f_i)\}$, Pattern $P = (s_o, f_o)$
2: **Output:** Maximum subset S' of mutually compatible activities that overlap with P
3: $F \leftarrow \{\}$
4: $counter \leftarrow 0$
5: **for** each activity (s_i, f_i) in S **do**
6: **if** _____(1)and_____ **then**
7: $F \leftarrow$ _____(2)_____
8: $counter \leftarrow counter + 1$
9: **end if**
10: **end for**
11: **if** counter > 0 **then**
12: F_____(3)_____
13: **else**
14: **return** F
15: **end if**
16: **if** F is not empty **then**
17: Sort the activities in F by finish time
18: $S' \leftarrow \{\}$
19: $A \leftarrow 0$
20: **for** _____(4)_____ **do**
21: **if** $s_i \geq A$ **then**
22: $S' \leftarrow$ _____(5)_____
23: _____(6)_____
24: **end if**
25: **end for**
26: **end if**
27: **return** S'

There are six 'fill-in the blanks' in the Algorithm 5 . Please write the response for 4.a.1, 4.a.2, 4.a.3, 4.a.4, 4.a.5, 4.5.6 respectively in separate lines. [6 marks]

(b) Compute the time-complexity of Algorithm 5. [2 Marks]

(c) For the input $S = \{(2,7),(3,5),(8,12),(1,4),(7,17)\}$ and the pattern $P = \{(3,5)\}$, compute the output returned by the Algorithm 5. [2 marks]

5. Understand the Algorithm 6 for the N-queen's problem and answer the following.

(a) In the Algorithm 6, first queen is placed in which row? [2 marks]

(b) If the board is of size $n \times m$, n >m, how many queens can be placed in the board satisfying all the constraints. [2 marks]

(c) How many diagonals are checked before placing a queen? [2 marks]

(d) If n=4, how many recursion calls are made? [2 marks]

(e) Compute the time-complexity of the recursive function alone in Algorithm 6. [2 marks]

Sample question template for the open-book examination

- All answers are to be written in a single line.

1. Given n matrices $A_1, A_2, ...A_n$, a **matrix-chain** of length n is a sequences of n matrices. For example, $< A_1A_2A_3A_4A_5 >$ is a matrix chain of length 5. Product of a matrix-chain $< A_1A_2, ..., A_n >$ is the product $A_1 * A_2 * A_3 * ... * A_n$. That is, Product of a matrix-chain is the product of all the matrices of the matrix-chain in the same order. Product of the matrix-chain $A_1A_3A_4$ is the product $A_1 * A_3 * A_4$. Given n matrices $A_1, A_2, A_3, A_4, ..., A_n$ of order $p_0 \times p_1$, $p_1 \times p_2, ..., p_{n-1} \times p_n$ respectively, design a dynamic programming based pseudocode to compute the parenthesization of the product of the chain-matrix of length n so that the number of scalar multiplications involved in the computation of the product of the chain-matrix is minimum

Algorithm 1 Matrix Chain Multiplication

(a)

```
1: Procedure MatrixChainMultiplication p, n
2: Input: Dimensions of matrices p[0...n]
3: Output: Minimum number of scalar multiplications
4: Let m[1...n][1...n] be a new table
5: Let s[1...n][1...n] be a new table
6: for i = 1 to n do
7:    m[i][i] = 0
8: end for
9: for l= 2 to n do
10:    for i = 1 to n − l + 1 do
11:       j = _____(1)_____
12:       m[i][j] = ∞
13:       for k = i to j − 1 do
14:          q = _____(2)_____
15:          if q < m[i][j] then
16:             _____(3)_____
17:             _____(4)_____
18:          end if
19:       end for
20:    end for
21: end for
22: return m and s
23: print PrintOptimalParens(s, 2, n)
24: Procedure PrintOptimalParens(s, i, j)
25: if _____(5)_____ then
26:    Print A_i
27: else
28:    Print '('
29:    PrintOptimalParens(s, i, s[i][j])
30:    _____(6)_____
31:    Print ')'
32: end if
33: EndProcedure
```

Algorithm 1 is designed to solve the Matrix chain Multiplication problem. There are six 'fill-in the blanks' in the Algorithm 1. Please write the response for 1.a.1, 1.a.2, 1.a.3, 1.a.4 , 1.a.5, 1.a.6 respectively in separate lines. [6 marks]

(b) If n=6, and $p = [2, 5, 7, 9, 11, 5, 6]$, Compute the the entry in the $[4, 5]^{th}$ entry in the m-table. [2 marks]

(c) For the same input as in 1(b), Compute the the entry in the $[5, 6]^{th}$ entry in the s-table. [2 marks]

Algorithm 2 PQRS

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

2. Understand the Algorithm 2 and answer the following.

(a) If the Algorithm 2 gives out 456, write the input fed into the Algorithm 2. [2 marks]

(b) If 23456 is given to the Algorithm 2, compute the output? [3 marks]

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

(d) Compute the recurrence relation that lies behind the Algorithm 2. [2 marks]

Reg. No.:

Name :

VIT

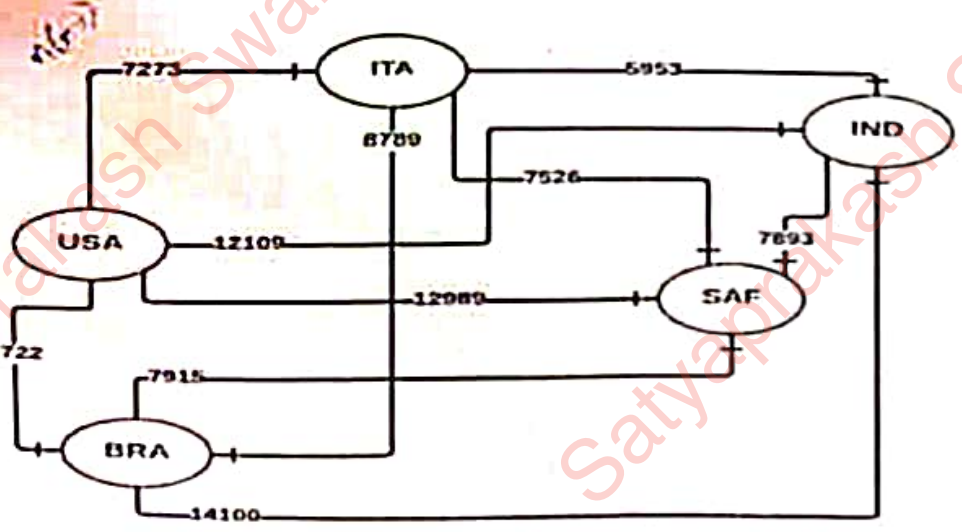
Vellore Institute of Technology

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

CAT – 2 March- April 2022

Programme	: B.Tech CSE	Semester	: Winter 21-22
Course Code	: CSE2012	Arrear Class ID	: CH2021225000708
Course Title	: Design and Analysis of Algorithms		
Faculty(s)	: Dr. Venkatraman S.		
Time	: 90 mins	Max. Marks	: 50

Answer all the Questions (5 X 10 Marks = 50 Marks)

1.		Given a chain <A1, A2, ..., An> of n matrices, where matrix Ai (i = 1, 2, ..., n) has the dimension P_{i-1} X P_i problem is to find the optimal sequence of pairings for multiplication of matrices A1, A2...An in a such way that the number of scalar multiplications required for the product from A1 to An is minimum. Construct an algorithm for matrix chain multiplication problem and illustrate your algorithm for A1*A2*A3*A4 and how it produces parenthesized product sequence in a way that minimizes the number of scalar multiplications. Where dimensions are A1 is 1x2 matrix A2 is 2x3 matrix A3 is 3x4 matrix A4 is 4x5 matrix	10 marks
2.		A travelling salesman plans to visit n cities. He wishes to visit each city only once, and again arriving back to his home city from where he started in such a way that the total travelling distance is minimum. Construct an algorithm for travelling salesman problem. Illustrate how your algorithm works for the below graph that consist of countries and the distances between each pair of countries. Find the shortest possible route that covers each country exactly once, starting from USA and returning back to the origin country?  	10 marks

3. Construct a backtracking algorithm to find all possible ways to place n/2 queens and n/2 Rooks(Elephants) on a n×n chessboard so that no two queens, no two rooks and no queen and rook attack each other.
Thus, a solution requires that

- no two queens share the same row, column, or diagonal,
- no two rooks share the same row or column
- no queen and rook share the same row, column, or diagonal.

Analyze your algorithm with time complexity. **10 marks**

4. Interwoven is a function which takes two strings S1 and S2 as input and generate a string S3 which is obtained by inserting characters of S2 into S1 in order. Few additional characters can be inserted into S1 to obtain S3.
For example, the strings S1= abac and S2= bbc occur interwoven in T = cabcbabcca.

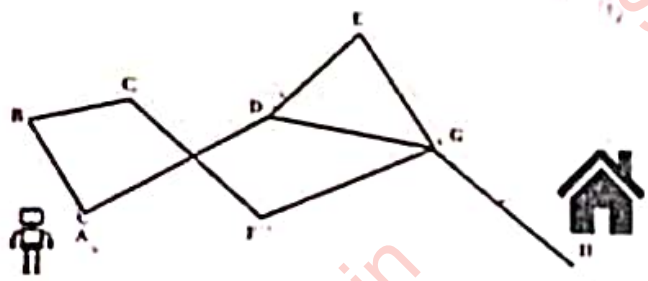
interwoven ("hello", "hai") = "heedhlalio"

h	c	e	d	h	l	a	l	i	o
S1	R	S1	R	S2	S1	S2	S1	S2	S1

S1 – Character from S1
R – Random character of user choice
S2 – Character from S2

Given two strings S1 and S2 and a text T, Design an algorithm to find whether there is an occurrence of S1 and S2 interwoven in T. **10 marks**

5. Consider a robot navigation problem where a robot is placed in a 2- dimensional environment, which has "n" line segments <L1,L2,L3,... Ln>, some of those lines are connected through end points.The task is to find all possible paths for robot from starting point of the line segment (L1) to end point of the line segment (Ln).



Construct an algorithm for robot navigation problem and illustrate how your algorithm works for the above diagram. (Identify all possible paths for a robot to travel from starting point A to reach the destination point H (House). (Note: Robot can turn left, trun right and move forward) **10 marks**
Example:
Input: AB, AD, BC, CF, DG, FG, DE, EG, GH
One of the possible path output:
Action taken by robot: Turn right and Move forward, Turn right and Move forward,Move forward
Path: AD, DG, GH

⇔⇔⇔

Reg. No.:

Name :

VIT

Vellore Institute of Technology

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

Continuous Assessment Test II – October 2022

Programme	:	M.Tech: CSE(BA)	Semester	:	FS 2022-23
Course	:	Design and Analysis of Algorithms	Code	:	CSE3037
			Class Nbr	:	CH2022231000982, CH2022231000983
Faculty	:	Dr. Ashoka Rajan R, Dr. Smrithy G S	Slot	:	A1+TA1
Time	:	90 Minutes	Max. Marks	:	50

Answer ALL the questions

Q.No.	Sub. Sec.	Questions	Marks
1.		A sequence is generated using a recurrence relation f as shown below: $f(n) = \begin{cases} f(n-1) + f(n-2) + f(n-3) & \text{if } n > 2 \\ 5 & \text{if } n = 0, 1, 2 \end{cases}$ Generate a sequence for n = 6 using the above recurrence relation. i. Design a dynamic programming (DP) algorithm to generate the sequence using above recurrence relation. **(6 Marks)** ii. How many times the function 'f' will be called in the DP algorithm designed when n=5? **(2 Marks)** iii. Analyse the time complexity of the designed DP algorithm. **(2 Marks)**	10
X 2.		Given two integer arrays profit [0..N-1] and weight[0..N-1] which represent profits and weights associated with N items respectively. Also given an integer W which represents knapsack capacity, find out the maximum profit subset of profit [] such that sum of the weights of this subset is smaller than or equal to W. You cannot break an item, either pick the complete item or don't pick it (0-1 property). Consider the following table where number of items N=4 and knapsack capacity W=40	10

items	1	2	3	4
weight	10	20	30	40
profit	30	10	40	20

Illustrate least cost branch and bound strategy to solve the above problem.

(10 marks)

3. In N queens problems we intend to place N queens on an N×N chessboard so that no two queens attack each other. A specialized version of the problem is columnized 6 queens where we place 6 queens on a 6×6 chessboard column wise. That is, first queen (Q1) can be placed on any row of 1st column, second queen (Q2) can be placed on any row of 2nd column, similarly Q3 in 3rd column, Q4 in 4th column, Q5 in 5^{th} column and Q6 in 6^{th} column without any attack.

Note: Queen can move diagonal, vertical or horizontal direction. **10**

(i) Design a backtracking algorithm to solve the given problem. **(4 Marks)**

(ii) Show all the possible safe queen positions for columnized 6 queens problem. **(4 Marks)**

(iii) Analyse the time complexity of the designed algorithm. **(2 Marks)**

4. Consider the following Text 'T' and pattern 'P' **10**

T : bacbababababacaca

P : ababaca

(i) Compute the longest prefix suffix function for the pattern P in T **(3 marks)**

(ii) Illustrate the step-by-step procedure of pattern (P) matching for the above text (T) using KMP algorithm. **(5 marks)**

(iii) Analyse the best case and worst case time complexity of KMP algorithm **(2 Marks)**

5. Consider the text T= "abccddaefg" and pattern P = "cdd". Let the alphabet size Σ = 10 (i.e. Base =10). Illustrate Rabin-Karp algorithm, **10**

i. To find the valid shift positions when you are scanning the text from left to right with modulo q=13. **(5 Marks)**

ii. Similarly, illustrate the valid shift positions when you are scanning the text from right to left without using the same modulo function. **(5 Marks)**

VIT

Vellore Institute of Technology

School of Computer Science and Engineering

Winter Semester 2023-2024

Continuous Assessment Test – II

SLOT: A2+TA2

Programme Name & Branch : B.Tech (BCB/BCE/BCI/BCT/BDS/BKT)

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

Class Number (s): ALL

Faculty Name (s): ALL

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

Answer All

Q.No.	Question	Max Marks
1.	A bank has 14 million dollars, which can be invested into stocks of four companies (1, 2, 3, and 4). The following table shows the net revenue of each company and the amount that must be invested into each company.	10

Company	1	2	3	4
Revenue (Million $)	16	22	12	8
Investment Amount (Million $)	5	7	4	3

The objective for the bank is to select a set of companies for investment, so as to maximize the total revenue with the condition that no partial investment can be done i.e., for each company we can either invest into it or not. Solve the problem with the algorithm whose exponential worst-case time complexity can be improved by employing better techniques for efficient pruning.

VIT

Vellore Institute of Technology

School of Computer Science and Engineering

Winter Semester 2023-2024

Continuous Assessment Test – II

Programme Name & Branch	: B.Tech – (BCB/BCE/BCI/BCT/BDS/BKT)	SLOT :A1+TA1
Course Name & code	: BCSE204L – Design and Analysis of Algorithms	
Class Number (s)	: ALL	
Faculty Name	: ALL	
Exam Duration	: 90 Min.	Maximum Marks: 50

ANSWER ALL THE QUESTIONS(5X10=50 Marks)

Q.No	Question	Max Mark
1	Solve the Knapsack Problem using FIFOBB, assume knapsack capacity is W =8. Show how queue is used for node creation in the state space tree. Item | Profit | Weight 1 | 13 | 4 2 | 15 | 2 3 | 14 | 4 4 | 16 | 6	10
2	Find the existence of a pattern P in the given string S (assign digits A-C as 0-2), using Rabin Karp algorithm. For hash function use Mod 13. Find out how many spurious hits does the algorithm encounter in the **Text =ABCBBCABCBAAABCCAACB** when looking for the pattern **Pattern = CCA**?	10
3	Consider a logistics manager tasked with optimizing transportation routes for a delivery company that operates in a city with a complex network of roads. Your goal is to minimize the distance for packages to reach their destinations by finding the shortest paths between all pairs	10

Registration Number : 22BCE1351

VIT
Vellore Institute of Technology
CHENNAI

Continuous Assessment Test II - October 2024

Programme	B.Tech.(CSE)	Semester	Fall 2024-25
Course	Design and Analysis of Algorithms	Code	BCSE 204L
Faculty	Dr.Srinivasa Rao U, Dr. Jayaram B, Dr.Tapabrata Roy, Dr. Sivapriya Dr. Lekshmi	Slot/Class No.	D2 + TD2 / CH2024250101207 /CH2024250101210 /CH2024250102302 /CH2024250101209 /CH2024250102301
Time	90 Minutes	Max. Marks	50

Instructions:

- Answer all 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: Time-complexity.

---

1. You are given a boolean expression E consisting of symbols from the alphabet Z ={**T**, **F**, ∧**(and)**,∨**(or)**, ⊕**(XOR)**}. The task is to count the number of ways to parenthesize the expression E such that it will evaluate to **True**. For example, there are 2 ways to parenthesize the expression $\mathbf{E} = \mathbf{T} \wedge \mathbf{F} \oplus \mathbf{T}$ as $(\mathbf{T} \wedge \mathbf{F}) \oplus \mathbf{T}$ and $\mathbf{T} \wedge (\mathbf{F} \oplus \mathbf{T})$ where both evaluate to **True**. Write an algorithm using a Dynamic Programming approach to count the number of ways to paranthesize the expression. Illustrate the algorithm with your own example and derive its time complexity. [15 marks]

[**Rubrics**: Logic: 3 marks, Illustration: 3 marks, Algorithm: 7 marks, Time-complexity: 2 marks]

2. Subset-sum problem: The Subset-sum problem is defined as follows: Given a set $A = [a_1, a_2, a_3, \ldots, a_n]$ of n positive integers and a positive integer d, find a subset of A whose sum is equal to d. For example, if $A = 1, 2, 5, 6, 8$ and $d = 9$, there are two possible solutions: $\{1, 2, 6\}$ and $\{1, 8\}$. Of course, some instances of this problem may have no solution. A backtracking algorithm for this problem is given below as **Algorithm 1**. The algorithm is missing some parts that are marked using blanks(– – – –). Your task is to fill these missing blanks in the algorithm. Illustrate the algorithm with your own example and derive the time complexity of the algorithm. [15 marks]

[**Rubrics**: Fill in the blanks: 8 marks, Illustration: 4 marks, Time-complexity: 3 marks]

Algorithm 1 SUBSET-SUM(A, d)

```
1: Input: Set A = [a_1, a_2, a_3, ..., a_n] of n positive integers and a positive integer d.
2: Output: A subset of A whose sum is equal to d.
3: result =[ ]
4: solutions =[ ]
5: BACKTRACK(start, currentsum)
6: if (- - - - - -) then
7:    solutions.append(copy(result))
8:    return
9: end if
10: if (- - - - - -) then
11:    return
12: end if
13: for i = start to len(A) - 1 do
14:    result.append(A[i]);
15:    BACKTRACK(- - - - - - - - - - - - - - - - - -);
16:    result.pop();
17: end for
```

3. Let $T[1 \ldots n]$ and $P[1 \ldots m]$ be a text of n characters and a pattern of m characters respectively taken from the alphabet $S = \{a, b, c, d, \ldots, z\}$. The pattern P is said to occur at shift s in T if $P[1 \ldots m] = T[s+1 \ldots s+m]$. Many a times it is possible that P does not occur in full but occurs partially in T. We are now interested in finding those partial occurrences of P in such a way that at least $2/3^{rd}$ of P occurs in T. In this context we define the following:

We say that a shift s is a ***fully valid shift***, if $P[1 \ldots m] = T[s+1 \ldots s+m]$. A shift s is said to be ***invalid shift***, if $P[1 \ldots k] \neq T[s+1 \ldots s+k]$ for any value of $k < 2m/3$. A shift s is said to be ***partially valid shift*** if $P[1 \ldots k] = T[s+1 \ldots s+k]$, for any value of k, $\frac{2m}{3} \leq k \leq m$. It is possible that there are many partially valid shifts s in T. Modify the Rabin-Karp algorithm to output all the partially valid shifts s along with the value of k for which $P[1 \ldots k] = T[s+1 \ldots s+k]$. [10 marks]

[**Rubrics**: Logic: 2 marks, Illustration: 3 mark, Algorithm: 3 marks and Time-complexity: 2 marks]

4. Consider the following variation of the Job Scheduling Problem - You are given two identical processors and n different jobs each with their processing times $p_1, p_2, p_3, \ldots p_n$. The task is to process the n jobs on either of the processors without pre-emption(an assigned job cannot be withdrawn from a processor before completion) so that the total time taken by both the processors to complete the n jobs is minimized. Write an algorithm that uses a greedy strategy to complete all the n jobs in minimum possible time. Illustrate your algorithm with an appropriate example. [10 marks]

[**Rubrics**: Logic: 2 marks, Illustrations: 3 mark, Algorithm: 3 marks and Time-complexity: 2 marks]

$\iff \iff \iff \iff$

VIT
Vellore Institute of Technology

Re-Continuous Assessment Test II – November 2024

Programme	B.Tech.(CSE)	Semester	Fall 2024–25
Course	Design and Analysis of Algorithms	Code	BCSE 204L
Faculty	Dr.Srinivasa Rao U, Dr. Jayaram B, Dr.Tapabrata Roy, Dr. Sivapriya Dr. Lekshmi	Slot/Class No.	D2 + TD2 / CH2024S0101207 /CH2024250101210 /CH2024250102302 /CH2024250101209 /CH2024250102301
Time	90 Minutes	Max. Marks	50

Instructions:
Answer all 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: Time-complexity.

You are a zookeeper in the reptile house. One of your rare lizards has just had several babies. Your job is to find a place to put each baby lizard in a nursery. However, there is a catch: the baby lizards have very long tongues.

A baby lizard can shoot out its tongue and eat any other baby lizard before you have time to save it. As such, you want to make sure that no baby lizard can eat another baby lizard in the nursery (burp).

For each baby lizard, you can place them in one spot on a grid. From there, they can shoot out their tongue up, down, left, right and diagonally as well. Their tongues are very long and can reach to the edge of the nursery from any location.

Figure 1 shows in what ways a baby lizard can eat another.

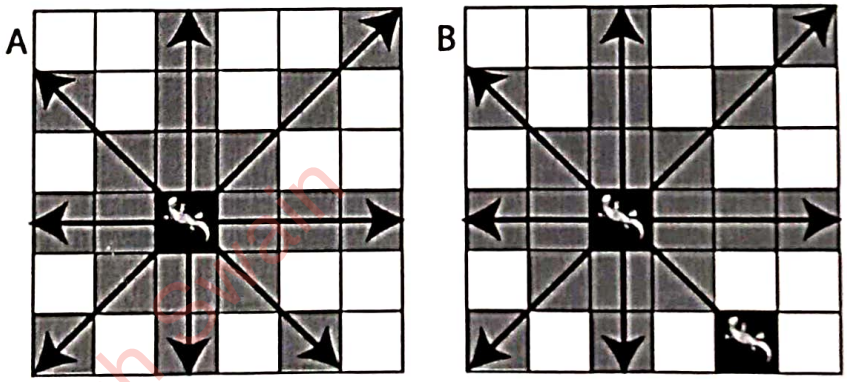


Figure 1 (A) the baby lizard can attack any other lizard in a red square. Thus it can be seen that a baby lizard can eat another lizard to its top, bottom, left right or diagonal. (B) In this example setup, both lizards can eat each other. Your algorithm will try to avoid this.

In addition to baby lizards, your nursery may have some trees planted in it. Your lizards cannot shoot their tongues through the trees nor can you move a lizard into the same place as a tree. As such, a tree will block any lizard from eating another lizard if it is in the path. Additionally, the tree will block you from moving the lizard to that location. Figure 2 shows some different valid arrangements of lizards:

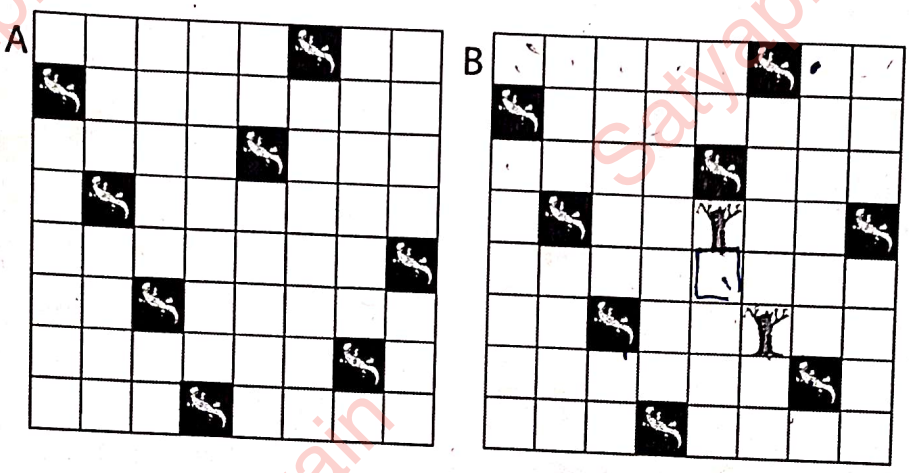


Figure 2 Both nurseries have valid arrangements of baby lizards such that they cannot eat one another. (A) with no trees, no lizard is in a position to eat another lizard. (B) Two trees are introduced such that the lizard in the last column cannot eat the lizard in the second or fourth column.

Given an arrangement of trees, we need to output a new arrangement of lizards such that no baby lizard can eat another one. You cannot move any of the trees. **[15 marks]**

[Rubrics: Logic: 3 marks, Illustration: 3 marks, Algorithm: 7 marks, Time-complexity: 2 marks]

2. Given a string, return the number of distinct substrings using Rabin Karp Algorithm. [10 marks]

Input : str = "aba"
Output : 5
Explanation :
Total number of distinct substring are 5 - "a", "ab", "aba", "b" ,"ba"
Input : str = "abcd"
Output : 10
Explanation :
Total number of distinct substring are 10 - "a", "ab", "abc", "abcd", "b", "bc", "bcd", "c", "cd", "d"

[Rubrics: Logic: 2 marks, Illustration: 3 mark, Algorithm: 3 marks and Time-complexity: 2 marks]

3. You are developing a scheduling system to maximize the number of online courses that students can complete within given time constraints. Each course is represented by a pair (t,d), where t is the number of consecutive days required to complete the course, and d is the last day by which the course must be completed. Courses must be taken one at a time, starting from day 1, and each course can only be added to the schedule if it is completed on or before its closing day, d. Your objective is to find the maximum number of courses that can be completed within these constraints. Given an integer n representing the number of courses, and an array of pairs courses where each pair (t,d) describes a course (with $1 \le t,d \le 10^4$), output

an integer representing the maximum number of courses that can be completed. For instance, if the input is 4 courses represented by [[100, 200], [200, 1300], [1000, 1250], [2000, 3200]], the output should be 3, as only three courses can be completed on time. Design an efficient solution using a **greedy technique** to ensure the highest number of courses can be taken, considering deadline order and selectively removing the longest course when the total schedule exceeds a deadline. **[10 marks]**

[Rubrics: Logic: 2 marks, Illustration: 3 mark, Algorithm: 3 marks and Time-complexity: 2 marks]

4. The Longest Increasing Subsequence (LIS) problem is to find the length of the longest subsequence of a given sequence such that all elements of the subsequence are sorted in increasing order. Write an algorithm using dynamic programming that determines the LIS of a string 'x'. For example, the length of LIS for {10, 22, 9, 33, 21, 50, 41, 60, 80} is 6 and LIS is {10, 22, 33, 50, 60, 80}. **[15 marks]**

Algorithm LIS(x)
Input: A sequence x of length n
Output: Length of the longest increasing subsequence in x

```
1: let    LIS = [1] * n          // Initialize an array to store LIS lengths for each position
2: for i = 1 to n - 1 do
3:     for j = 0 to i - 1 do
4:         if _____ and _____ then
5:             _____ = LIS[j] + 1
6:         end if
7:     end for
8: end for
9: return _____                   // Maximum value in LIS array is the length of the
longest increasing subsequence
```

[Rubrics: Fill in the blanks: 8 marks, Illustration: 4 marks, Time-complexity: 3 marks]

***** All The Best *****

Let's Connect.....!! 😍

VIT-C 26 (Satya Helpzz)
VIT-C 27 (Satya Helpzz)
VIT-C 28 (Satya Helpzz)

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

Subscribe to my Youtube channel & Join my groups now ! Don't miss out! ⏭ 🚀

Know me As a Person - Whatsapp Channel And here is my MasterLink 🔗 (Everything i do)

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

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

Youtube Instagram LinkedIn Facebook