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	G1+TG1
		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

Q1. 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)

Q2. 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]

(iii) Describe the functionality of the algorithm F? [2 Marks]

(iii) What is the time-complexity of the algorithm F? [2 Marks]

(iv) For which input value, algorithm F_1 returns the same value [2 Marks]

(v) Describe the functionality of the algorithm F, if line: 14 in the algorithm F_1 is deleted? [2 Marks]

63. Consider a string X. We define Single-Circular-Shift, written as SCS(X), which is obtained by moving the character in the first position of X to the last position of X. For example SCS(ABCDE)= BCDEA . Also $SCS^2(X)$ = SCS(SCS(X)) i.e SCS^2(ABCDE)=CDEAB Similarly, we define $SSC^k(X)$, where k<length(X). We define Half-Circular-Shift of X, written as HCS(X), as the SCS $^{\lfloor n/2 \rfloor}$ (X), where $\lfloor n/2 \rfloor$ is the usual floor operator. HCS(pqrs)=rspq Given a text T, pattern P, of length n and m respectively (n>m). Design a hash-based pseudocode to compute all the valid shifts of the occurrence of HCS(P) in T. For example given T: theadmissiblenetrkwo, P: work, your algorithm should output 16 shifts. [10]

Rubrics:

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

64. a. Consider a Problem P_1: Given a set A of n integers, the task is to count the number of integers whose multiples are also in A itself. If A={5,10,15,23,46} solution to problem P_1 is 2, since the integers 5 & 23 have their multiples in A itself. Identify the class-complexity (P NP NPC) of the problem P_1 and justify your answer. [10]

b. Set A is said to be a subset of a set B if all elements of A are also elements of B. Consider a problem P_2: Given a set of integers S, your task is to split the set S into two subsets S_1 and S_2 such that the sum of elements in S_1 is equal to the sum of elements in S_2 and return -1 if set S cannot be split into two sets S_1 and S_2 satisfying the above constraint. For example, if S = {5, 6, 11} then S_1 = {5,6} S_2 = {11}.

Identify the class- complexity (P / NP / NPC) of the problem P_2 and justify your answer.

Rubrics:

a. Identification of class-complexity (2 marks), Justification (3 marks)

b. Identification of class-complexity (2 marks), Justification (3 marks)

Section B (4 X 15 Marks)

Answer all questions

65. Consider the matrix_chain $<A_1 * A_2 * \ldots A_n>$ of n matrices whose dimensions are $d_0 \times d_1$, $d_1 \times d_2$, $d_2 \times d_3$, $\ldots d_{n-1} \times d_n$ respectively. Given an array d with the elements d_0 d_1 $\ldots$ d_{n-1} d_n, design a dynamic programming pseudocode to compute the minimum number of scalar multiplication required to compute the product of the chain $((A_1 * A_2) * A_3) * (A_4 * A_5 * \ldots A_{n-1}) * (A_{n-2} * (A_{n-1} * A_n))$ Please note that you are not supposed to change the parenthesization involving A1, A2, A3, and the parenthesizations involving A_{n-2},A_{n-1},A_n while calculating the minimum number of scalar multiplication required to compute the above product. [15]

Rubrics:

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

66. The convex-hull of a set of Q points, denoted by CH(Q) is the smallest convex polygon P for which each point in Q is either on the boundary of P or inside P. Given a set of points Q and another point p, design an algorithm to construct the convex hull CH(Q) and determine whether p lies inside CH(Q) or on the boundary of CH(Q), or outside CH(Q). [15]

Rubrics:

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

07. Every cell in a n x n grid, G is represented by a pair of positive integers [i, j] which conveys the usual meaning. Neighboring cell of a cell 'C' in G consists of cells that are above, below, left and right of 'C'. In other words, the neighbor cells of ([i, j]) are the cells: {[i-1, j], [i+1, j], [i, j-1], [i, j+1]}. G, can be filled with white and black coins. A white coin in the cell [i,j] is said to be locked if all of its neighboring cells contain black coins, and similarly, a black coin in the cell [i, j] is said to be locked if all of its neighboring cells contain white coins. Given a n x n grid G, where n is an even number, design a backtracking algorithm to count the number of ways to fill G with n^2 coins(half of the n^2 coins are white and half of the coins are black) in such a way that no coins in the grid are locked. [15]

Rubrics:

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

08. Consider the network G = (V, E, C, s, t) where V is the set of vertices, E is the set of edges, C is the capacity of the edges, s is a vertex designated as source vertex, t is a vertex designated as sink vertex. Design an algorithm to compute the maximum flow in G, written as Max_flow(G). Choose any vertex $v \in V$ such that $v \notin \{s,t\}$ and construct two networks G_1 & G_2 from the given network G as follows: [15]

Construction of network G_1: s is the source vertex of G_1, v is the sink vertex of G_1. G_1 will not have any outgoing edges from v and all the capacities of the edges in G_1 will be the capacities of the respective edges in G.

Construction of network G_2: v is the source vertex of G_2, t is the sink vertex of G_2. G_2 will not have any incoming edges to v and all the capacities of the edges in G_2 will be the capacities of the respective edges in G.

Thus from the network, G=(V, E, C, s, t), we have got two new networks G_1=(V, E_1, C_1, s, v) where $E_1 \subset E$ is got by deleting all the outgoing edges of v, C_1 gives the capacities, edges E_1 as in G.

G_2= (V, E_2, C_2, v, t) where $E_2 \subset E$ is got by deleting all the incoming edges of v, C_2 gives the capacities of edges E_2 as in G.

Let First_half_Flow(G) = Max_Flow(G_1) and Second_Half_Flow(G)=Max_Flow(G_2). Given G=(V, E, C, s, t) design an algorithm to verify the statement "Max_Flow(G)= First_Half_Flow(G)+Second_Half_Flow(G)". Your pseudocode should output true if the statement is valid, else it should return false.

Rubrics:

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

ᔕᔕᔕ