Sardar Patel Institute of Technology

Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058, India
(Autonomous College Affiliated to University of Mumbai)

End Semester Examination

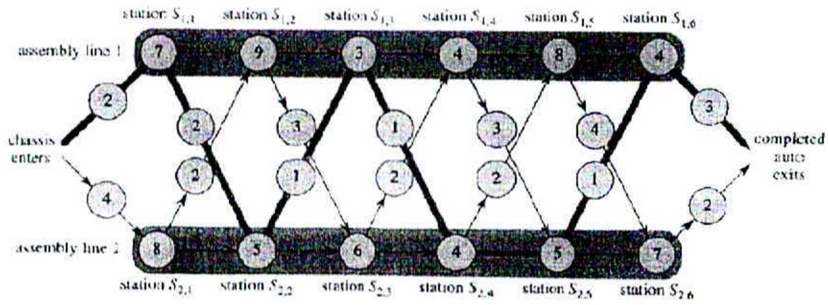
May-2023

Max. Marks: 100 **Duration: 180 Mins**
Class: B.Tech. **Semester: IV**
Course Code: CS501 **Branch: Computer Engineering/AIML/DS**
Name of the Course: Design and Analysis of Algorithms

Instruction:

(1) All questions are compulsory
(2) Draw neat diagrams
(3) Assume suitable data, if necessary

Q No.		Max. Marks	CO
Q.1(A)	Prove that the running time of an algorithm is θ(g(n)) if and only if its worst case running time is O(g(n)) and its best case running time is Ω(g(n)).	04	CO1
Q.1 (B)	Although Merge Sort runs in θ(nlogn) worst case time and insertion sort runs in $\theta(n^2)$ worst case time, the constant factors in insertion sort can make it faster for small problem sizes on many machines. Thus, it makes sense to use insertion sort within merge sort when subproblems become sufficiently small. Consider a modification to merge sort in which n/k sublists of length k are sorted using insertion sort and then merged using the standard merging mechanism, where k is a value to be determined. a. Show that insertion sort can sort the n/k sublists, each of length k, in θ(nk) worst case time. b. Show how to merge the sublists in θ(nlg(n/k)) worst case time OR Write a Binary Search Algorithm using recursive approach. Derive it's time complexity? Find the element 37 in array A[1,3,5,7,11,13,17,19,23,29,31,37,41,43,47,53,59] using Binary recursive Search:	08	CO2

Q.1(C)	Prove the master theorem: Let $a \geq 1$ and $b > 1$ be constants, and let f(n) be a nonnegative function defined on exact power of b. Define T(n) on exact power of b by the recurrence $$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ aT(n/b) + f(n) & \text{if } n = b^i \end{cases}$$ where i is a positive integer. Then $$T(n) = \Theta(n^{\log_b a}) + \sum_{j=0}^{\log_b n-1} a^j f(n/b^j)$$	08	CO1
Q.2(A)	What is accounting method? Illustrate accounting method amortized analysis using the example of stack operations.	05	CO2
Q.2(B)	a. What are the sequence of different steps need to follow in Dynamic Programming? b. Write an recursive algorithm to compute a car fastest way through the factory using Assembly Line Scheduling. c. The problem is to determine which station to choose from Line-1 and which to choose from Line-2 in order to minimize total time through the factory for one Auto using four steps of Assembly-Line-Scheduling. 	15	CO3
Q.3(A)	Write algorithm for job sequencing with deadlines. What is the solution generated using the algorithm when $n = 7, (p_1, p_2, ...p_7) = (3, 5, 20, 18, 1, 6, 30), and (d_1, d_2, ..., d_7) = (1, 3, 4, 3, 2, 1, 2)$	10	CO3
Q.3(B)	Write algorithm for greedy strategies for the knapsack problem. Find an optimal solution using greedy strategy to the knapsack instance $n = 7$, $m = 15$, $(p_1, p_2, ...p_7) = (10, 5, 15, 7, 6, 18, 3)$, and $(w_1, w_2, ..., w_7) = (2, 3, 5, 7, 1, 4, 1)$	10	CO3
Q.4(A)	Let $n = 6, m = 30, w = \{5, 10, 12, 13, 15, 18\}$ and $m = 30$. Find all possible subsets of w that sum to m using the algorithm sum of subsets. Draw the state space tree that is generated.	10	CO4

<table>
<tr><td>Q.4(B)</td><td>Apply LC Branch and bound technique to solve the 15-puzzle problem for an initial arrangement given below.
<table><tr><td>1</td><td>3</td><td>4</td><td>15</td></tr><tr><td>2</td><td></td><td>5</td><td>12</td></tr><tr><td>7</td><td>6</td><td>11</td><td>14</td></tr><tr><td>8</td><td>9</td><td>10</td><td>13</td></tr></table></td><td>10</td><td>CO4</td></tr>
<tr><td>Q.5(A)</td><td>Illustrate with example an approximation algorithm for traveling salesman problem showing the optimal cost and the approximate cost of the tour.</td><td>10</td><td>CO4</td></tr>
<tr><td>Q.5(B)</td><td>Illustrate Rabin Karp string matching algorithm? working modulo $q = 11$, how many spurious hits does the Rabin-Karp matcher encounter in the text $T = 31415922653589793$ when looking for the pattern $P = 26$?</td><td>10</td><td>CO5</td></tr>
</table>