



**AUTUMN END SEMESTER EXAMINATION-2022**

**V Semester, B.Tech (Course)**

**DESIGN & ANALYSIS OF ALGORITHM CS-2012**

**(For 2020 Admitted Batches)**

**Time: 3 Hours Full Marks: 50**

***Answer any SIX questions.***

***Question paper consists of four SECTIONS i.e. A, B, C and D.***

***Section A is compulsory.***

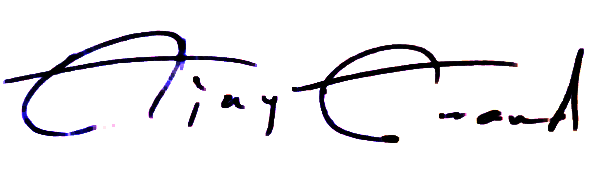
***Attempt minimum one question each from Sections B, C, D.***

*The figures in the margin indicate full marks.*

*Candidates are required to give their answers in their own words as far as practicable*

*and all parts of a question should be answered at one place only.*

| **SECTION-A (**Learning levels 1 and 2) | | | | | Learning levels as per Bloom’s taxonomy | Course Outcomes (CO) |
| --- | --- | --- | --- | --- | --- | --- |
| 1. |  | Answer the following questions. | [110] | | 1 | 2,3 |
|  | (a) | When is Greedy Strategy preferred over Dynamic Programming? |  | | 1 | 1 |
|  | (b) | ***The worst case performance is used as a benchmark in analyzing algorithms***. Support or Contradict the statement with valid argument. |  | | 2 | 2,3 |
|  | (c) | Presence of Optimal Substructure is an essential criteria for the application of Dynamic Programming Paradigm. Support or Contradict the statement with valid argument. |  | | 2 | 3 |
|  | (d) | Does this graph have all source destination shortest paths? Support your answer. |  | | 3 | 1,2 |
|  | (e) | What is the key cost that is associated , in choosing Greedy over Dynamic Programming design paradigm? |  | | 2 | 1,2 |
|  | (f) | Consider the following Code:  N = input from user  I = 1;  J = 1;  for(i = 1; i<N; i++)  {  for(j = i; j >= 1; j = j /2)  {  print(“\--Poverty Eradication through Education--/”);  }  }  Find out the Time complexity of the program above and also establish a relation between line number and number of times the message prints in each line. |  | | 2 | 6 |
|  | (g) | For problem X, we can design a Non Deterministic Turing machine which is able to solve X in polynomial Time. But for some instances the said Turing Machine does not halt. Write something about the complexity class of Problem X. |  | | 1 | 4 |
|  | (h) | Differentiate between a Substring and Subsequence through appropriate examples. |  | | 2 | 5 |
|  | (i) | Using different types of Data structure for intermediate storage for the Graph Traversal yields different traversal. Suggest modification to a Traversal algorithm such that it can be modified to a Minimum Spanning Tree Algorithm. |  | | 2 | 5 |
|  | (j) | Why asymptotic analysis is preferred over exact analysis. Give two key points. |  | | 1 | 1 |
|  |  |  |  | |  |  |
| **SECTION-B (**Learning levels 1,2, and 3) | | | | | Learning levels as per Bloom’s taxonomy | Course Outcomes (CO) |
| 2. | (a) | Considering a Comparison operation consumes 1 unit of CPU time and a swap operation takes that of 3: identify the better sorting algorithm to check if an array is already sorted or not, among Insertion sort and selection sort. (No marks will be given if you write pseudocode) . You may use an array of size 5 (not more) to verify and support your answer |  | [4] | 1,3 | 1,3 |
|  | (b) | Use Prim’s Algorithm to find a Minimum spanning Tree for the graph shown below. Use the Node at the 12 o'clock to start. |  | [4] | 2,3 | 3,4 |
|  |  |  |  |  |  |  |
| 3. | (a) | Consider a matrix chain . The dimensions of these matrices are p0 = 4, p1 = 5, p3 = 6 , p4 = 5, p5 = 4. Find the optimal cost through the number of scalar multiplication for this chain. |  | [4] | 1,3 | 2, 3,4 |
|  | (b) | To the above matrix chain, find the optimal parenthesization |  | [4] | 1,3 | 2, 3,4 |
|  |  |  | |  |  |  |
| **SECTION-C (**Learning Levels 3 and 4) | | | | | Learning levels as per Bloom’s taxonomy | Course Outcomes (CO) |
| 4. | (a) | Write down equivalent iterative routine for the Recursive Routine , whose performance can be depicted through this set of recurrence relation: |  | [4] | 3 | 1,2 |
|  | (b) | Consider the randomized version of the Quicksort. Due to some issue with the Pseudo Random number generator , only 1/3rd of the pivots generated are in the optimal range. Would it still be preferable to you instead of the normal Quicksort. Support or contradict your answer through an equation. |  | [4] | 4 | 3,4 |
|  |  |  |  |  |  |  |
| 5. | (a) | Find out the length of Longest Common Subsequence between  X : **BHUBANESWAR** and Y : **BERHAMPUR**  (No marks will be given for writing Pseudo-codes) |  | [4] | 3.4 | 3.4 |
|  | (b) | For the above X and Y : fetch the Longest Common Subsequence. |  | [4] | 3,4 | 3,4 |
|  |  |  |  |  |  |  |
| 6. | (a) | “*BFS is a shortest path algorithm in some special cases*”. Do you agree or not? If you support this answer, do specify the special case. |  | [4] | 3 | 3,4 |
|  | (b) | Find the shortest path from Node A to all other Node for the following |  | [4] | 4 | 3,4 |
|  |  |  | |  |  |  |
| **SECTION-D (**Learning levels 4,5,6) | | | | | Learning levels as per Bloom’s taxonomy | Course Outcomes (CO) |
| 7. | (a) | A problem can be solved using recursion. For each function call made, subsequently 3 recursive calls are made, with problem size reduced to 1/4th of the original problem size. Upon solving these problems the solutions can be combined, in quadratic with respect to problem size. The trivial problem can be solved for the problem size 1, and the cost associated with it is 0. Formulate the recurrence relation. |  | [4] | 4,6 | 2,3 |
|  | (b) | Solve the above equation using recurrence tree method. And give your comment on the work done at each level of the recurrence tree. |  | [4] | 4, 5 | 1,2 |
|  |  |  |  |  |  |  |
| 8. | (a) | Inversion Count for an array indicates – how far (or close) the array is from being sorted. If the array is already sorted, then the inversion count is 0, but if the array is sorted in reverse order, the inversion count is the maximum.  Modify any known sorting algorithm to **find the number of inversion pairs** in an array. |  | [4] | 5 | 1,2,3 |
|  | (b) | Rahul suggested that he solved a problem named C, using Divide and Conquer strategy. The performance of the algorithm is described by recurrence relation followed. Earlier the Brute Force approach had a time complexity in factorial of the Problem Size. Comment on the Combine Phase of the algorithm & how it is affecting the overall performance.  The Recurrence Relation:  T(n)=2T(n/3)+cn2  T(1)=0 | | [4] | 5 | 1,2,3 |
|  |  | \*\*\*\*\* | |  |  |  |

* *

***Signature of Paper Setter/Moderator***