2022

COMPUTER SCIENCE
(Practical)

Paper : CSMP : 105

[Module – A]

(Data Structures and Algorithms)

Full Marks : 50

Follow the instructions carefully :

- **Candidates are required to answer in their own codes and words as far as applicable.**
- **The answer script should have a top page (Page #1) as an index page; mention page number(s) against the answer of each part of the question(s).**

Instructions :

1. **Marks Distribution :**

   a. **Sessional + Laboratory Notebook + Attendance : 15+5+5 = 25**

   b. **Examination + Viva voce : 20+5 = 25**

      (i) **Algorithm : 3**

      (ii) **Data structure(s) and its justification : 3**

      (iii) **Analysis Algorithmic complexity : 3**

      (iv) **Example / Dry run : 4**

      (v) **Result : 5**

      (vi) **Special features / Discussions : 2**

2. **The examiner will distribute the questions from the set of questions by a lottery system with a minimum number of repetitions of questions / problems.**

S(1st Sm.)-Computer Sc.-CSMP : 105(Module-A)/(Pr.)

PROBLEM SET

1. Devise a static scheme in computing a polynomial 'C' where 'C' is computed by

   (a) Multiplying two polynomials A and B, and

   (b) Differentiating polynomial A. Clearly mention the assumptions, if any, and declare your achievements. 14+6

2. Determine the *biconnected components* of undirected graphs each of order at least 15. Choose appropriate data structure(s) and algorithmic technique(s) for the said purpose and make clear why and how the data structure(s) is (are) suitable. 20

3. Relatively compare graph representations using Incidence Matrices and Adjacency Lists for the following types of graphs; clearly mention the assumptions, if any.

   (a) Undirected unweighted graph,

   (b) Undirected weighted graph,

   (c) Directed unweighted graph, and

   (d) Directed weighted graph. 5+5+5+5

4. Employ *disjoint set data structures* to compute the components as well as the number of components and degree of each of the vertices of a given graph *G* of order at least 15. Also, compute the pendant vertices, isolated vertices, and maximal cliques present in *G*, if any. Choose appropriate data structure(s) for the said purpose and make clear why and how the data structure(s) is (are) suitable. 20

5. (a) Consider the following graphs and employ Depth First Search algorithm on them to compute the Depth First Search trees: (i) Undirected graph, and (ii) Directed graph.

   Clearly mention the assumptions, if any, and declare your achievements.

   (b) Employ the same search technique to compute the number of components of a given undirected graph. 12+8

6. Consider sequences each of at least 20 elements and perform *SieveSort* for each of them. Choose appropriate data structure(s) for the said purpose and make clear why and how the data structure(s) is (are) suitable. 20

7. (a) Consider the following graphs and employ Breadth First Search algorithm on them to compute the Breadth First Search trees : (i) Undirected graph and (ii) Directed graph.
Clearly mention the assumptions, if any, and declare your achievements.

(b) Employ the same search technique to compute the number of components of a given undirected graph. 12+8

8. Devise a static scheme in computing a polynomial 'C' where 'C' is computed by (a) Adding two polynomials A and B, and (b) Subtracting polynomial B from polynomial A.
Clearly mention the assumptions, if any, and declare your achievements. 10+10

9. Devise a scheme to represent a sparse matrix X of order at least 10, and transpose this representation of X in lexicographic order. Choose appropriate data structure(s) for the said purpose and make clear why and how the data structure(s) is (are) suitable. 20

10. Relatively compare graph representations using Incidence and Adjacency Matrices for the following types of graphs; clearly mention the assumptions, if any.

(a) Undirected unweighted graph,

(b) Undirected weighted graph,

(c) Directed unweighted graph, and

(d) Directed weighted graph. 5+5+5+5

11. Consider a sequence of at least 15 elements and construct an AVL tree after insertion of each element into the tree. Also, perform deletion of at least five elements from the AVL tree, and rebuild the AVL tree as needed. Choose appropriate data structure(s) for the said purpose and make clear why and how the data structure(s) is (are) suitable. 20

12. Consider a directed as well as an undirected graph and devise a suitable search mechanism on each of them to compute the shortest distance and the path between a pair of vertices, if one exists. Clearly mention the assumptions, if any, and declare your achievements. 20

13. Let A be an $n \times n$ matrix that contains non-zero elements present only in five diagonals, centring and including the major diagonal. If the elements in the band formed by these five diagonals are represented column wise in an array B, with A[1, 1] being stored in B[l], obtain an algorithm to determine the value of A[i, j], $1 \leq i, j \leq n$ from array B. Choose appropriate data structure(s) for the said purpose, and make clear why and how the data structure(s) is (are) suitable. 20

14. Consider sequences each of at least 20 elements and perform *GraphSort* for each of them. Choose appropriate data structure(s) for the said purpose and make clear why and how the data structure(s) is (are) suitable. 20

15. Devise a dynamic scheme in computing a polynomial 'C' where 'C' is computed by

(a) Multiplying two polynomials A and B, and

(b) Differentiating polynomial A.

Clearly mention the assumptions, if any, and declare your achievements. 14+6

16. Consider directed graphs of order and size not less than 15 each, and employ Depth First Search algorithm on each of them; differentiate their edges based on your sequence of visiting the vertices. Clearly mention the assumptions and achievements, if any. 20

17. Consider the problem of incrementing n-digit binary integers and compute its amortized cost a_i in terms of n, if applicable. Choose appropriate data structure(s) for the said purpose and make clear why and how the data structure(s) is (are) suitable. 20

18. Relatively compare graph representations using Adjacency Matrices and Adjacency Lists for the following types of graphs; clearly mention the assumptions, if any.

(a) Undirected unweighted graph,

(b) Undirected weighted graph,

(c) Directed unweighted graph, and

(d) Directed weighted graph. 5+5+5+5

19. Perform the operation of *Binomial_Heap_Extract_Min* for a Binomial heap that comprises at least six Binomial trees, where the minimum key (value) is present in a node that is belonging to the largest tree or the next to the largest tree. Choose appropriate data structure(s) for the said purpose and make clear why and how the data structure(s) is (are) suitable. 20

20. Determine the strongly connected components of directed graphs each of order at least 15. Choose appropriate data structure(s) and algorithmic technique(s) for the said purpose and make clear why and how the data structure(s) is (are) suitable. 20

S(1st Sm.)-Computer Sc.-CSMP : 105(Module-A)/(Pr.)

21. Consider undirected graphs of order and size not less than 15 each, and employ Depth First Search algorithm on each of them; differentiate their edges based on your sequence of visiting the vertices. Clearly mention the assumptions and achievements, if any. 20

22. Devise a dynamic scheme in computing a polynomial 'C' where 'C' is computed by

(a) Adding two polynomials A and B, and

(b) Subtracting polynomial B from polynomial A.

Clearly mention the assumptions, if any, and declare your achievements. 10+10

23. Let X and Y be two lower triangular matrices, each with n rows. Devise a scheme to represent both the triangles in an array Z of size $n \times (n+1)$. Write algorithms to determine the values of $X[i, j]$ and $Y[i, j]$, $1 \le i, j \le n$ from array Z. Choose appropriate data structure(s) for the said purpose, and make clear why and how the data structure(s) is (are) suitable. 20

24. Write an algorithm that computes a B-tree of order m and perform insertion of an element into the tree as well as deletion of an element from the tree. As for example, consider the keys to be inserted are: $a_5, g_2, f_1, b_3, k_3, d_6, g_4, m_5, j_2, e_3, s_1, j_1, r_5, z_7, b_6, m_2, m_9, t_3, u_5, q_6$ into an initially empty tree, in the order given, when the value of $m = 5$. Also delete the keys one after another, from the final B-tree obtained, in the following order : g_4, r_5, q_6, d_6, and obtain subsequent B-trees. 14+6