MCACT202 Data structures and Algorithm Analysis

Question Bank

Module I

Part A

1.Explain about Data structures.
2.Write a note on various categories of data structures with suitable examples.
3.List any three advantages of Data structures.
4.What are linear Data structures.Write examples.
5.What are Non linear Data Structures.Write examples.
6.What are the basic operations performed on Data structures.
7. What is an Algorithm? Why are they important?
8.Explain Algorithm analysis in short.
9.write a note on Asymptotic notations.
10.Write an Algorithm to find the average of n numbers.
11.Calculate Time complexity for adding n numbers.
12.What are the characteristics of an Algorithm?
13.write a note on Array.
14.Illustrate multidimensional arrays.
15.Compare Arrays and Linked List.
16.Explain disadvantages of Arrays.
17.How can we overcome the limitations of Array?
18.write a note on Stack.
19.Explain any three applications of Stack.
20.What are the basic operations performed on Stacks.
21.Explain Stack a)using Arrays b)using Linked list.
22.Explain role of Stack in Recursive algorithms.
23.Write a note on Queue data structure.
24.Write anote on different types of Queues.
25.What do you mean by Priority Queue.
26.What are the advantages of Priority queue over normal queues?
27.Explain any one application of Priority queue.
28.Convert infix expression (a+b)*c into postfix expression.
29.Convert infix expression (a+b)/(c*d) into prefix expressions.
30.Why prefix and postfix are more efficient than Infix in Algorithms and arithmetic operations.

Part B

1) Write an algorithm to convert Infix expression to postfix using stack. Explain with an example.

2) Explain the different operations performed on a circular queue with algorithms and example.

3) Write an algorithm to perform polynomial addition using arrays.

4)Explain different operations performed on Stack.

5) Explain Space complexity of an Algorithm with examples.

6)Explain Time complexity of an Algorithm with examples.

7) Explain different operations performed on Queue with suitable algorithms and example.

8) Explain asymptotic complexity with suitable example.

Module II

Part A

1. What is a Linked list?
2. How will you represent a linked list?
3. With suitable example write short note on Sparse matrices
4. Write an algorithm to insert a node at the begging of a linked list
5. Write an algorithm to inserting a node at the end of a linked list
6. Write an algorithm to deleting the a node from a linked list
7. Explain doubly linked list with example
8. Define tree with example
9. Summarize the node,level and degree of a tree
10. With the help of an example explain Height and depth of a tree
11. What is a complete binary tree
12. What is a full binary tree
13. Explain Tree traversal with example

14. Define a binary search tree

15. What is balanced factor of a node

16. State a digraph?

17. What is weighted graph

18. What is an isolated vertex

19. Define the degree of vertex in a graph

Part B

1. With an algorithm explain the linked list insertion

2. Explain polynomial addition using linked list

3. Explain stack operation using linked list

4. With suitable example explain tree traversal

5. With algorithm explain insertion of Doubly linked list

6. Explain Binary tree, Balanced binary tree and AVL tree with example

7. Explain tree traversal with suitable example and algorithm

8. Explain Singly linked list deletion with algorithm

9. Explain binary search trees(BST)

10. With an algorithm explain DFS

11. Explain BFS with suitable example and also write an algorithm

12. Explain the Tree terminologies

Module III

Part A

1) Define the searching process in an array.
2) What is linear search?
3) What is binary search?
4) Compare linear and binary search methods.
5) Explain the complexities of linear and binary search.

6) Demonstrate how to perform insertion sort?
7) Define the time complexity for insertion sort.
8) Write the procedure for performing selection sort.
9) Prove the function for comparison in selection sort F(n)=O(n 2)
10) Different sorting examples
11) What is radix sort?
12) Define Heap?
13) Compare max heap and min heap?
14) Demonstrate the steps to do heap sort.
15) Summarize the worst case and best case analysis of radix sort.
16) Define hashing and hash functions.
17) Generate how truncation method is used for choosing a hash function.
18) Apply mid square method for the given set of keys
19) Compare the folding method and modular method.
20) Formulate how the hash function is used for floating point numbers.
21) What is meant by open hashing?
22) Explain linear probing method of collision resolution.
23) Compare open hashing and closed hashing methods.
24) State the advantages of quadratic probing method.
25) How a separate chaining method is implemented.

Part B

1. Explain how binary search is different from linear search with suitable example.
2. Explain the different methods for choosing hash functions.
3. Illustrate how the closed hashing is used to resolve collisions?
4. List out the selection sort procedure with suitable examples.
5. Explain heap sort with an example.
6. Explain insertion sort with example
7. When do we use hash functions? And describe any 4 hash functions.
8. Analyze radix sort with its time complexity.
9. Illustrate how the hash functions for floating point numbers and strings can be found with examples.
10. Evaluate the procedure of performing heap sort.
11. Compare open hashing and closed hashing methods.

Module IV

Part A

1. Divide and conquer method of algorithm design

2. Recurrence form of complexity for many Divide and conquer algorithms.

3. Straight forward method of finding the maximum and minimum in a set of n elements.

4. Computing time for finding Maximum and minimum using Divide and conquer.

5. Importance of divide and conquer method of algorithm design.

6. How Divide and conquer strategy is applied to Binary search.

7. Time complexity of Binary search

8. Computing time of Binary search in the case of best, average, and worst cases in successful and unsuccessful searches

9. Explain Quick sort.

10. Quick sort is considered to be the best sorting method. Why?

11. Explain Merge sort.

12. Merging of two lists

13. Divide and conquer strategy applied to Merge sort.

14. Describe Greedy method.

15. Differentiate Divide and conquer method and Greedy method

16. Greedy method in solving the knapsack problem.

17. State the knapsack problem

18. Feasible and Optimal solutions.

19. Spanning tree with Example

20. Minimum cost spanning trees and its applications

21. Method of generating minimum cost spanning tree generated from a graph using Prim's algorithm?

22. Method of generating minimum cost spanning tree generated from a graph using Kruskal's algorithm?

23. Difference between Prim's and Kruskal's algorithms

24. Stages of forming the minimum cost spanning tree of the graph given using Prim's algorithm (graph to be given)

25. Stages of forming the minimum cost spanning tree of the graph given using Kruskal's algorithm (graph to be given)

Part B

1. Divide and Conquer strategy with its control abstraction
2. Divide and conquer method for finding the maximum and minimum in a set of n elements
3. Analysis of Binary search algorithm
4. Algorithm for Quick sort.
5. Algorithm for Merge sort
6. Greedy method with its control abstraction
7. Greedy algorithm for knapsack problem.
8. Give an instance of the knapsack problem like {n= , m= , (p1,p2,…) = (……) and (w1,w2,…) = (……) . Find an optimal solution.
9. Kruskal's algorithm with example.
10. Prim's algorithm with example

Module V

Part A

1. Explain the concept of dynamic programming
2. Briefly explain the properties of dynamic programming
3. Explain the concept used in multistage graphs
4. Explain all pairs shortest path
5. Explain shortest path algorithm and its time complexity
6. What do you mean by the term principle of optimality
7. How can we implement dynamic programming
8. What do you mean by the term back tracking
9. Explain 8-Queens problem
10. How can we apply backtracking in N-queens problem
11. List out the various applications of back tracking
12. How can we represent the solution of a N -Queens problem
13. How can we determine the efficiency of a backtracking algorithm?
14. Explain branch and bound
15. Explain the different types of search in branch and bound
16. Explain the concept of live node and dead node
17. Explain the concept of lower bound theory
18. Explain LC search with an example
19. Explain the control abstraction of LC search
20. Briefly explain the concept used in LC search
21. Explain the technique used in decision trees
22. Explain ordered searching and sorting
23. Which method is used in comparison trees? Explain

24. Draw a de decision tree for sorting 3 numbers

Part B

1.Explain dynamic programming technique with an example
2.Explain the concept of multistage graph
3.Explain all pairs shortest path problem with an example
4.Explain backtracking with suitable example
5.Explain 8-Queens problem
6.Explain the various searching techniques used in Branch and Bound method
7.Explain LC Search with an example
8.Explain control abstraction for LC search
9.Explain comparison trees for searching and sorting.
10.Explain lower bound theory