Fundamentals of Data Structures and Abstract Data Types

1. “To write an efficient program, we should know about data structures.” Explain the above statement.
2. Explain the use of Big O notation in analyzing algorithms. Compare sorting time efficiencies of Quick-Sort and Merge-Sort.
3. Write Short notes on: ADT
4. Justify the statement: “To write an efficient program, we should know about data structures and algorithms”.
5. What is Big ‘O’ notation? Analyze any one sorting algorithm.
6. What is an algorithm? What is to analyze in algorithm? Define Big C = Oh notation for time complexity measurement of algorithm.
7. What is dynamic memory allocation? How it is achieved for declaring low dimensional array? Explain.
8. Write short notes on (any two):

b) ADT

1. How to find complexity of algorithms? Explain.
2. How to find complexity of algorithms? Explain.
3. What is an algorithm? Write down the features of an algorithm.
4. What do you mean by complexity of algorithm? How do you find time complexity?
5. Write short notes on: Dynamic memory allocation
6. How do you find complexity of algorithms? Explain
7. Why do we need asymptotic notation? Describe about Big oh notation with its curve.
8. What is Big Oh Notation?
9. What is the worst-case time complexity of quick sort? Sort the following data using quick sort algorithm: 20, 50, 45, 36, 8, 90, 85, 34

Linear Data Structures:

1. Explain the implementation of stack and queue with example.
2. How can you convert from infix to post fix notation?
3. How can you use Queue as ADT?
4. Write a menu program to demonstrate the simulation of stack operations in array implementation.
5. State relative merits and demerits of contiguous list and Linked list. Explain the steps involved in inserting and deleting a mode in singly linked list.
6. Write C function to display all the items in a circular queue in array implementation. Write assumptions, you need.
7. Explain CLL, DLL, DCLL (Circular, Doubly, Doubly Circular Linked List).
8. Define stack as ADT. Describe its primitive operations on Array implementation and linked list implementation.
9. Differentiate between Singly linked list, DLL, CLL and DCLL.

Describe circular Queue operations in array implementation.

1. Discuss the merits and demerits of contiguous list and linked list.
2. What is priority queue? How it is best implemented?
3. Define Queue as an ADT. Write a program for basic operations in Linear queue in array implementation.
4. Differentiate between contiguous list and linked list with examples.
5. Explain why linked list is called dynamic list? Write the algorithm for deleting a new node before a node.
6. Define Queue as ADT. Describe its primitive operation on array implementation and linked list implementation.
7. Explain the algorithms for infix to postfix conversion and evaluation of postfix expression. Trace the algorithms with suitable example.
8. How do you insert a nodes at last in doubly linked list? Explain.
9. State relative merits & demerits of contiguous list and linked list.
10. Trace out Infix to Postfix conversion algorithm with given Infix expression.  
    A + (((B-C) \* (D-E) + F)/G) $ (H-I)  
    Evaluate the postfix expression acquired from above for the given values:  
    A = 6, B = 2, C = 4, D = 3, E = 8, F = 2, G = 3, H = 5, I = 1.
11. Explain the structure of Doubly Linked List (DLL). Differentiate the difference between DLL and Doubly Circular Linked List (DCLL). Explain the procedures to insert a node in DLL at the beginning and at the last.
12. Write C function to insert an item circular queue in array implementation. Write assumptions, you need.
13. Discuss array as an ADT.
14. Write short notes on (any two):

a) Queue in circular linked list

1. What is stack? How is it different from queue? Write a program to implement all stack operations.
2. What is linked list? Explain the process of inserting and removing nodes from a linked list.
3. Transform the postfix expression AB − C + DEF − + $ to infix.
4. Write short notes on:

b. Circular queue

1. What is stack? How is it different from queue? Write a program to implement all stack operations.
2. What is linked list? Explain the process of inserting and removing nodes from a linked list.
3. Discuss array as an ADT.
4. Transform the postfix expression AB − C + DEF − + $ to infix.
5. What is Postfix expression? Write an algorithm to evaluate value of postfix expression. Trace the following expression into postfix expression.  
   **(A+B\*C) D E/ F) + −**
6. What is circular queue? Write an algorithm and C function to implement Circular queue.
7. How stack as ADT? Explain with example.
8. Write an algorithm and C function to delete node in singly link list.
9. Write short notes on:

b. Doubly Link list

How can you use stack to convert an infix expression to postfix? Convert infix expression **(A+B)\*(C-D)** to postfix using stack.

Compare stack with queue. How is linear queue different from circular queue?

What is ADT? Discuss stack as an ADT.

What are benifits of using linked list over array? How can you insert a node in a singly linked list?

Define queue. What are different applications of queue? Explain queue operations with example.

Explain circular linked list with example. How do you implement linked list operation in singly linked list? Explain

Evaluate the expression **ABCD-x+** using stack where A=5, B=4, C=3 and D=7.

What is priority queue? Why so we need this type of queue?

Explain array implementation of lists.

How recursive algorithm use stack to store intermediate results? Illustrate with an example. Convert the infix expression **A+B-(C\*D/E+F)-G\*H** into postfix expression using stack.

How do you insert and delete a node at kth position of the doubly linked list? Describe the process of implementing stack and queue using linked list. .

Define a queue. Explain about enqueue and dequeue operation in circular queue.

Evaluate the postfix expression **574-\*8/4+** using stack.

Write short notes on:

* a. Priority Queue

Write an algorithm for Enqueue operation.

What is a priority queue?

Convert the given infix expression ((a-b)(x+y/f-ce+d) into postfix using stack method.

Trees:

1. What do you mean by binary tree? Explain the binary search tree with example.
2. A binary tree T has 12 nodes. The in-order and pre-order traversals of T yield the following sequence of nodes:
   1. In-order : VPNAQRSOKBTM
   2. Pre-order : SPVQNARTOKBM
3. Construct the Binary tree T showing each step. Explain, how you can arrive at solution in brief?
4. Describe properties of Binary Search Tree. Write recursive algorithms for constructing BST and its traversals. Illustrate them with an example.
5. Construct the Binary tree T showing each step. Explain, how you can arrive at solution in brief?
6. Construct an expression tree from the following postfix:  
   **AB + C\*DC – -FG + $**
7. Write the steps involved in deleting a node in a Binary selection tree.
8. Describe the significance of Huffman tree. Describe procedure for construction of a Huffman tree. Illustrate it with example. Describe different types of applications of Binary trees.
9. Write about applications of Binary trees.
10. Define Binary Search Type (BST). Write an algorithm to insert a node in non-empty BST. Construct BST from the data:  
    **10, 20, 30, 25, 27, 7, 4, 23, 26, 21**
11. A screenshot of a computer

    Description automatically generated
12. Explain almost complete binary tree with example.
13. Write short notes on:

a. Tree traversal

1. Explain almost complete binary tree with example.
2. What is binary search tree? Explain with an example. Write an algorithm to search, insert and delete node in binary search tree.
3. How do you traverse a binary tree? Discuss.
4. What is binary search tree? Write a program to implement insertion and deletion algorithm in binary search tree.
5. Write short notes on: (2.5+2.5)

b. AVL tree

1. Why do we need to balance the binary search tree? Justify with example. Create an AVL tree from the data **24, 12, 8, 15, 35, 30, 57, 40, 45, 78**.
2. What is a Binary Tree?
3. What is the advantage of AVL tree over binary search tree? Construct an AVL tree from the given set of data50, 20, 60, 10, 8, 15, 32, 46, 11, 48.

. Indexing Methods:

Algorithm Design Techniques:

1. What do you mean by recursion? Explain the implementation of factorial and Fibonacci sequences with example.
2. Explain Divide and Conquer algorithm taking reference to Merge Sort.
3. Write recursive algorithm to get Fibonacci term. Illustrate it drawing recursion tree.
4. Why recursion is required? Explain with Tower-of-Hanoi example. How recursive algorithm makes program effective? Write the merits and demerits of recursion in Programming.
5. Write merits and demerits of recursive function over non-recursive function.
6. State TOH problem. Write recursion tree when no. of disks are four.
7. State TOH problem. Explain a recursive algorithm to solve the problem.
8. What is recursion? Write a recursive program to find factorial of a number.
9. What is recursion? Write a recursive program to find factorial of a number.
10. What is Recursion? Write a recursive algorithm to implement binary search.
11. Explain concept of divide and conquer algorithm. Hand test quick sort algorithm with array of numbers (78, 34, 21, 43, 77, 18, 9, 56, 38, 19). What is time complexity of quick sort algorithm?
12. Define recursive algorithm? How do you implement recursive algorithm while writing computer programs?
13. Write a recursive program to find nth Fibonacci number.
14. Write short notes on: (2.5+2.5)

a. Divide and conquer sorting

1. What is recursion?
2. What is a greedy algorithm?
3. Write short notes on: a. Divide and conquer b. Dynamics programming
4. What are the major characteristics of algorithms?

Graph Algorithms:

1. What is Post-order traversal?
2. Differentiate between Pre-order and In order traversal.
3. Describe strong and weekly connected graphs with examples. What is weighted graph?
4. Differentiate between tree and graph. What are spanning forest and spanning tree. Explain MST (Minimum cost Spanning Tree) problem.
5. State MST (Minimum Cost Spanning Tree) problem and shortest path (single source and all other destination) problem. Name the algorithms for solving these problems.
6. Explain Kruskal’s algorithm with example.
7. What is weighted graph? Explain Depth-first traversal of a graph.
8. Write short notes on (any two):

c) MST (Minimum Cost Spanning Tree) of a graph.

1. What is graph traversal? Discuss depth-first traversal technique with suitable example.
2. What is graph traversal? Discuss depth-first traversal technique with suitable example.
3. What do you mean by graph traversal? Explain primes algorithm with example.
4. Discuss depth first and breadth first traversal of a graph with suitable example.
5. What is graph traversal? Explain.
6. A diagram of a graph using prim's

   Description automatically generated

Write short notes on:

b. Breadth First Traversal of a graph.

A diagram of a triangle with circles and letters

Description automatically generated

What is a graph?

Explain Dijkstra shortest path algorithm. Calculate the shortest path of the following graph using Dijkstra algorithm.

Searching, Merging, and Sorting:

1. What is sorting? Describe the Insertion.
2. Explain the binary searching.
3. Trace Binary Search algorithm for the data: **21, 36, 56, 79, 101, 123, 142, 203**. And Search for the values 123 and 153.
4. What are external and internal sorting? Explain partition strategies of Merge sort and Quick sort. Trace these sort algorithms for following data:  
   **11 45 61 33 55 9 83 25**
5. Compare and Contrast between Binary searching and Binary tree searching.
6. Write Short notes on: Hash function
7. State collision resolution techniques in hashing. Explain double hashing and quadratic probing techniques.
8. Write a program in C for bubble sorting.
9. Explain binary search. Illustrate it with example.
10. Explain hashing with example.
11. Explain the characteristics of Huffman’s algorithm and its application.
12. Discuss merge sort. How you rate this sorting from selection sort?
13. Compare partition strategies of Merge sort and Quick sort.
14. Explain Bubble sort algorithm. Illustrate it with an example.
15. Describe recursive procedure of Binary searching technique? Discuss about efficiency of Binary searching.
16. What are Hashing and collision? Write about any three hashing algorithms
17. Trace selection – sort algorithm for the following data:  
    **42, 23, 74, 11, 65, 58, 94, 86**
18. What is Hashing? What collision means? State collision resolution techniques. Explain one of them in brief.
19. Explain efficiency of
    1. a) Binary Searching
    2. b) Quick sort
20. Write a program to sort an array using selection sort.
21. Discuss binary search technique along with its efficiency.
22. Why do we need Hashing? Discuss linear probing in detail.
23. Hand test the insertion sort algorithm with following array of numbers.  
    **16 7 31 2 9 41 -10**
24. Write a program to sort an array using selection sort.
25. Discuss binary search technique along with its efficiency.
26. Why do we need Hashing? Discuss linear probing in detail.
27. Write an algorithm and C function for merge sort.
28. Differentiate between selection sort and bubble sort.
29. Write short notes on:

a. Hashing

How do you implement binary search algorithm? What is time complexity of this algorithm?

What is hashing? Discuss rehashing with example

Hand test selection sort with array of numbers **4, 71, 32, 19, 61, 2, -5** in descending order.

Write a program to implement sequential search algorithm.

Write a program to implement binary search.

1. In which case, the position od pivot element in quick sort is always either in first or last position. Create a max heap from the numbers {10, 12, 53, 34, 23, 77, 59, 66, 5, 8}. (2+3)

What is hash function?

What is a internal sorting?

 Define max and min heap. Construct a binary max heap using following set of data and illustrate the deletion operation on this heap: 11,2,9,13,57,25,17,1,90,3

