Chapter 1 : Introduction to Data Structure

Question Bank

(Answers of following Questions are available in the form of e-book)

1.1 Introduction to Algorithms

Q. Define algorithm? **(May 2011)**

Q. Discuss the parameters on which efficiency of an algorithm is defined.

Q. What are different symbols used in flowchart? Explain with diagram.

Q. Write an algorithm for adding 10 numbers. Draw the flowchart for same.

Q. What are the different algorithm design tools?

1.2 Data

Q. Define data?

Q. Define following terms with respect to data structure:

1. Atomic data 2. Composite data.

1.3 Data Types

Q. Define term data type.

Q. What are the different data types we use in C language? Explain any 4 with example.

Q. Define terms: 1. Structure 2. Union.

1.4 Abstract Data Types (ADT)

Q. What does abstract data type mean? **(Dec. 10)**

Q. Explain the term data object.

1.5 Data Structures

Q. Define data structure. List the various linear and non-linear data structures and explain them in brief. Also state the difference between them. **(Dec. 09, May 10, May 12, Dec. 11, Dec. 13, May 13)**

Q. Differentiate the following terms :
Primitive and non-primitive data structures **(May 11)**

Q. Write a short note on persistent and ephemeral data structure.

Q. Write a short note on static and dynamic data structure.

Q. Explain static and dynamic memory allocation techniques.

1.6 Relation between Data Structure and Algorithm

Q. Explain relation between data structure and algorithm with the example.

1.7 Algorithm Analysis

Q. Define time and space complexity of an algorithm. **(March 10)**

Example 1.7.7 : Greatest Common Divisor (GCD) is defined as :

$$GCD(n, m) = \begin{cases} GCD(m, n) & , \text{if } m < n \\ m & , \text{if } n \bmod m = 0 \\ GCD(m, n, \bmod m) & , \text{otherwise} \end{cases}$$

Calculate GCD (3,16)

What are the time and space requirement of your algorithm ?

Example 1.7.9 : Define Time complexity and Space complexity. Calculate time complexity for given expression.

```
for (k=0; k<n; k++)
{
    rows[k] = 0;
    for(j=0; j<n; j++)
    {
        rows[k] = rows[k] + matrix[k][j];
        total = total + matrix[k][j];
    }
}
```

(May 12)

Q. Determine worst case time complexity of the following program segment.

```
for i = 1 to n
   for j = 1 to n
for k = 1 to j
   a = 1;
end;
    end;
end;
```

Q. Write down algorithm for addition of two numbers and find out its time complexity.

Q. What do you mean by frequency count ? State its complexity.

1.8 Different Algorithm Asymptotic Notation

Q. Explain the following terms with its graphical representation.

1. Big oh (O)
2. Omega (Ω)
3. Theta (θ)

Q. Explain the term Big oh (O) with particular example.

Q. Explain the term Omega (Ω) with particular example.

Q. Explain the term Theta (θ) with particular example.

Q. Explain the term Small oh (o) with particular example.

Q. Write a short note on Order of Growth.

Example 1.8.2 : Which function grows faster ?

(i) $n^{\log n}$; $(\log (n))^n$ (ii) $\log_n k$; $(\log n)^k$

(iii) $n^{\log \log \log n}$; $(\log n)!$ (iv) n^n ; $n!$

Q. Explain concept of best case, worst case and average case behaviour of an algorithm with particular example.

Chapter 2 : Arrays

Question Bank

(Answers of following Questions are available in the form of e-book)

2.1 Sequential Organization

Q. Explain the term sequential organization with its properties.

Q. Write array as an ADT.

2.2 Introduction to Arrays

Q. Explain the term array with example.

2.3 Representation and Analysis

Q. Explain how we represent and analyze the array with example.

Program 2.3.1 : Write a program to display array elements with their addresses.

2.4 One-Dimensional Arrays

Q. Explain the term 1-D array with example.

Q. What is the term traversing of linear array ?

2.5 Operations with Arrays

Q. What the different operations possible with the array?

Q. How we perform merging operation of two sorted arrays? Explain with example.

Q. Write pseudo 'C' code to reverse the 'n' numbers in one dimensional array.

2.6 Two-Dimensional Arrays

Q. Write short note on : Storage representation of 2 Dimensional array.

(Dec. 2011)

Q. Write a short note on address calculation of the 2 D array.

Example 2.6.1 : Each element of an array Data [20][50] requires 4 bytes of storage. Base address of Data is 2000. Determine the location of Data [10][10] when the array is stored as : (i) Row major. (ii) Column major.

2.7 Multi-Dimensional Arrays

Example 2.7.1 : Consider integer array, int arr[5][4] declared in 'c' program. If the base address is 510, find the address of the element arr[3][2], with row major and column major representation of array.

Example 2.7.2 : Given a two dimensional array Z1 (2 : 9, 9 : 18) stored in column major order with base address 100 and size of each element is 4 bytes. Find address of the element Z1 (4, 12). **(May 2012)**

Q. Explain multidimensional array. How it is stored in memory? **(Dec. 2013)**

2.8 Application of Arrays

Q. Write an algorithm to multiply two matrices. Also perform time analysis for the same. **(May 2011)**

Q. Write a C function which obtains the transpose of an n × n square matrix on to itself.

Program 2.8.1 : Write a program for addition of two matrices.

Program 2.8.3 : Write a program for checking of symmetry.

Program 2.8.4 : Write a program for multiplication of two matrices.

Q. Write C function to determine the location of saddle point if exists.

2.9 Character String in C-Language

Program 2.9.1 : Write a program for usage of string handling functions.

Program 2.9.2 : Write a program for find length of string.

Program 2.9.3 : Write a program for copy string.

Program 2.9.4 : Write a program for comparing two strings.

Program 2.9.5 : Write a program for reversing the given string.

Program 2.9.7 : Write a program for concatenates the two strings.

Q. Write a program for finding the substrings from given string.

Q. Explain the term character string in C language.

2.10 Pointers in C

Q. Write a short note on pointers in C language.

Program 2.10.1 : Write a program to interchange contents of two variables x and y. Variables x and y are accessed through pointers.

Q. List out the advantages of pointers in C language.

Program 2.10.2 : Write a program to find the sum of the elements of an array using pointers.

2.11 Dynamic Memory Allocation

Q. Write a short note on dynamic memory allocation.

Program 2.11.1 : Write a program to concatenate two strings. Program should use a function for concatenation and the function should return the final string.

Q. Write an algorithm for traversing an 1 – D array and generating address of each of its element.

Q. List out advantages of dynamic memory allocation over static memory allocation.

Chapter 3 : Standard Data Structures

Question Bank

(Answers of following Questions are available in the form of e-book)

3.1 Ordered List

Q. Write a function in any programming language to insert an element in an ordered list. **(May 11)**

Q. Explain the term ordered list with particular example.

Q. Write down 'C' function to add two polynomials.

Q. Write down 'C' function to multiply two polynomials.

Q. Explain how we represent the polynomial by making use of array.

3.2 Sparse Matrix as an Ordered List

Q. What is sparse matrix? Explain the representation of sparse matrix. **(March 10, Dec. 10, May 12)**

Program 3.2.1 : Write a program to implement transposition of a sparse matrix in C language.

Q. Draw the vector representation of the sparse matrix of the following matrix.

$$\begin{bmatrix} 0 & 12 & 0 & 0 \\ 0 & 11 & 0 & 0 \\ 9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Q. Write an algorithm to add two sparse matrices. How much time does your algorithm take?

Q. Draw the vector representation of the sparse matrix of the following matrix.

$$\begin{bmatrix} 0 & 0 & 11 & 0 \\ 12 & 0 & 0 & 0 \\ 0 & -4 & 0 & 0 \\ 0 & 0 & 0 & -25 \end{bmatrix}$$

Q. Explain term fast transpose with suitable example.

Example 3.2.1 : Represent the polynomial $5 + 10x + 5xy + 9x^2y^2 + 2x^3y$ using sparse matrix.

3.3 Introduction to Recursion

Q. What is recursion ? What care should be taken in writing recursive function ? Give example of any one recursive function. Write a C program for GCD using recursion. **(Dec. 09, May 11, Dec. 11)**

Q. Give difference between recursion and iteration. **(Dec. 10)**

Q. What is Tower of Hanoi? Explain it with n=3. **(May 12)**

Q. Explain following terms :

1. Backtracking
2. Removal of recursion
3. Tail recursion.

Q. How we use stack for function call? Explain with suitable example.

Q. A recursive function f is shown below. What is the value of f(5) ?

```
int f (int x)
    {
        if (x < 2)
            return (1);
        else
            return f(x + 1) + f (x – 2);

    }
```

3.4 Introduction to Stack

Q. Write an algorithm to convert infix expression to postfix expression. **(Dec 09, May 14)**

Example 3.4.1 : Write an algorithm for push, pop and empty operations on stack. Using above functions write an algorithm to determine if an input character string is of the form :

$a^i b^i$ where I > = 1 **(March 10)**

Example 3.4.2 : Write an algorithm which will check that the given string belongs to following grammar or not.

$L = \{wcw^R \mid w \in \{a, b\}^*\}$ (Where w^R is the reverse of w) **(Dec. 10)**

Example 3.4.4 : Consider the following arithmetic expression P, written in postfix notation. Translate it in infix notation and evaluate.

P : 12, 7, 3, –, /, 2, 1, 5, +, *, + **(Dec. 09)**

Example 3.4.5 : Write an algorithm for evaluation of postfix expression and evaluate the following expression showing status of stack in tabular form.

(i) 5 4 6 + * 4 9 3 / + * (ii) 7 5 2 + * 4 11 + / – **(Dec. 09)**

Example 3.4.6 : Find value of following postfix expression using stack trace.

3 5 * 6 2 / + **(Dec. 12)**

Example 3.4.7 : Evaluate the following postfix expression using stack

AB+CD/*GH*+ ((where A=2, B=4, C=6, D=3, G=8, H=7)

(May 14)

Example 3.4.9 : Evaluate following prefix expression.

+ * A B – C + C * B A (A= 4, B=8,C=12) **(May 12)**

Example 3.4.11 : Convert following infix expression to the postfix expressions. Show stack trace.

(i) A/B $ C + D* E/F – G + H

(ii) (A + B) * D + E/(F + G* D) + C **(May 13)**

Example 3.4.12 : Trace the conversion of infix to postfix form in tabular form.

(i) (A + B * C / D – E + F / G / (H + I)) **(Dec. 09)**

Q. Write the implementation procedure of basic primitive operations of the stack using linear array. Write an algorithm to implement PUSH, POP and CHANGE operations on stack **(Dec. 09, Dec. 11, Dec. 13)**

Example 3.4.14 : Convert following Infix expression into Postfix expression. Show each step.

A + B ^ C^ D - E * F / G **(Dec. 13)**

Example 3.4.15 : Translate the following string into polish notation and trace the content of stack

A - (B / C + (D % E * F) / G) * H **(May 14)**

Example 3.4.16 : Translate the following string into Polish notation and trace the content of stack

(a + b ^ c ^ d) * (e + f / d) **(Dec. 10)**

Example 3.4.18 : Transform the following expression to postfix and evaluate postfix expression by assuming A = 1, B = 2, C = 3, D = 4, E = 6, F = 6, G = 1, I = 3 and J = 3.

A + B – C * D / E + F $ G / (I + J) **(March 10)**

Example 3.4.29 : Convert following infix expression into postfix and prefix expression.

a – b / c * d + e * f / g **(Dec. 11)**

Example 3.4.30 : Convert the following string into prefix :

A-B/(C*D^E) **(Dec. 10)**

Example 3.4.31 : Write an algorithm to change the i^{th} value of stack to value X. **(Dec. 10)**

Q. Write an algorithm to convert parenthesized infix expression to postfix. **(May 11, May 12)**

Q. Write short note on : Evaluation of postfix expression using stack. **(Dec. 11)**

Q. What are the applications or uses of the stack? **(May 12, May 14)**

Q. Write an algorithm of postfix expression evaluation. **(May 13)**

Q. Write an algorithm to evaluate an arithmetic expression using stack and show how the expression 3 * (5 – 3) will be evaluate.

Q. Give the data structure to implement two stacks in same array. Write functions to implement Push operation on both the stacks.

Q. Explain the terms overflow and underflow with suitable example.

Q. How we perform conversion of an expression from infix to prefix? Explain with suitable example.

Q. How we perform conversion of an expression from postfix to infix? Explain with suitable example.

Q. How we perform conversion of an expression from postfix to prefix? Explain with suitable example.

3.5 Introduction to Queue

Q. Explain following :

(i) Priority queue (i) Circular queue **(Dec. 09)**

Q. Write the implementation procedure of basic primitive operations of the queue using linear array **(Dec. 09)**

Program 3.5.2 : Write a program to perform insert and delete operations on a circular queue. (May 11)

Example 3.5.3 : Consider the following queue, where queue is a circular queue having 6 memory cells.

Front = 2, Rear = 4.

Queue : _, A, C, D,_,_

Describe queue as following operation take place :
F is added to the queue
Two letters are deleted
R is added to the queue
S is added to the queue
One letter is deleted **(Dec. 10)**

Q. Give the difference between simple queue and circular queue. **(Dec. 11, Dec .13)**

Q. What do you mean by FIFO and LIFO ? **(Dec. 11)**

Q. Write an algorithm/program to implement delete operation into a circular queue using array representation of queue **(Dec. 11, Dec. 13)**

Q. Write short note on : Priority queue. **(Dec. 11)**

Q. Write short note on : Dequeue. **(Dec. 11)**

Q. Differentiate between stack & queue. Also explain priority queue. **(Dec. 09, May 12)**

Q. Write an algorithm for circular queue that insert an element at rear end. **(May 12)**

Q. Write an algorithm for double ended queue that insert an element at front end. **(May 12)**

Q. Explain insert and delete function of circular queue. **(Dec. 12)**

Q. Write insert and remove functions for queue if it is implemented using circular link list. **(May 13)**

Q. What are priority queues? Explain its uses. **(May 14)**

Example 3.5.4 : Consider a dequeue given below which has LEFT=1, RIGHT=5

_ A B C D E _ _ _ _ .

Now perform the following operations on the dequeue

1. Add F on the left.
2. Add G on the right.
3. Add H on the right.
4. Delete two alphabets from left
5. Add I on the right

Differentiate peep() and pop() functions **(May 14)**

Q. List the uses of queue. **(May 14)**

Chapter 4 : Linked Lists

Question Bank

(Answers of following Questions are available in the form of e-book)

4.1 Representation and Implementation of Singly Linked Lists

Q. Discuss advantages and disadvantages of stack and queue implemented using linked list over array. **(Dec. 09, March 10, Dec.11)**

Example 4.1.1 : Write an advantage of link list, doubly link list and circular link list. **(Dec. 10, May 11)**

Q. Differentiate between overflow and underflow condition in the linked list.

4.2 Basic Linked List Operations

Q. Write an algorithm to insert an element into a singly link list. **(Dec. 10)**

Q. Write a program to count number of nodes in a linked list. **(May 11)**

Q. Write an algorithm to delete an element from a singly linked list. **(May 11)**

Q. Write an algorithm/program to implement following operations in the "Singly Linked List" (i) Insert the node at end (ii) Delete the node whose value = Y. **(Dec.11, Dec. 12)**

Q. Write a C function search (I, x) that accepts a pointer 1 to a list of integers and returns a pointer to a node containing x if it exists and the null pointer otherwise. **(May 13)**

Q. Write an algorithm to insert a node in an ordered linked list. **(May 14)**

Q. Write an algorithm/program to "insert a node at end" operation of singly linked list. **(Dec.13)**

Q. Discuss implementation of singly linked list. Write C functions to implement following operations on singly linked list : (i) To count number of nodes (ii) To reverse the direction of the links. (iii) To delete alternate nodes that is first, third, fifth and so on.

Q. Write an algorithm to search an item from a sorted linked list.

Q. Write an algorithm to delete the last node from a singly linked list.

Q. Write an algorithm to concatenate two linked list.

Q. Explain merging of sorted linked list with suitable example

Q. Write down C program to create a single linked list and split it at the middle and make the second half as the first and vice versa. Display the final list.

Q. Write down C function for removing duplicate elements from a linked list.

4.3 Circular Linked List

Q. Write an algorithm to perform each of the following operations on circular singly linked list using header node.

1. Add node at the end.
2. Add node at the beginning.

Q. Write an algorithm for inserting and deleting an element into circular linked list. **(March 10, May 12)**

Q. Write short note on : Circular linked list. **(Dec. 10, Dec.11)**

Program 4.3.1 : Write a program to perform all (create, insert, delete, display) the operations in a circular linked list. **(May 11)**

Q. Write algorithm for searching an item in unsorted circular linked list.

Q. Write an algorithm that will concatenate two circular linked lists, producing a circular linked list.

4.4 Doubly Linked List

Q. Write an algorithm to delete an element from a doubly link list. **(Dec.10, Dec.13)**

Q. Write a short note on doubly link list. **(Dec. 10)**

Program 4.4.2 :. Write a program to perform all (create, insert, delete, display) the operations in a doubly linked list.

Q. Write an algorithm to insert and delete a node in doubly linked list. **Dec.11)**

Q. Write difference between singly linked list and doubly linked list. **(May 12)**

Q. Explain delete operation of doubly linked list. **(Dec.12)**

Q. Write a program in any programming language to concatenate two doubly linked lists. **(May 14)**

4.5 Doubly Linked Circular List

Q. Explain delete operation of doubly circular linked list.

Q. Write a short note on doubly circular link list

Q. Write a program to perform all (create, insert, delete, display) the operations in a doubly circular linked list.

Q. Write an algorithm to insert a new node into orderly doubly circular linked list.

Q. Write difference between singly linked list and doubly circular linked list.

4.6 Applications of Linked Lists

Example 4.6.1 : Write a C/C++ program to add two polynomials represented using doubly linear linked list. Also write necessary functions to represent polynomial using doubly linear link list. **(Mar. 10)**

Q. Discuss how list data structures are useful to represent a polynomial and performing various operations upon a polynomial.

Q. Write down C function for addition of two polynomials using linked list.

Q. Write down C function for multiplication of two polynomials using linked list.

4.7 Concept of Skip List

Q. Write a short note on skip list.

4.8 Storage Management

Q. Explain different ways of storage management.

4.10 Operations on Queue Implemented using Linked Structure

Q. Write the implementation procedure of basic primitive operations of the Queue using linked list. **(Dec. 09)**

Q. Write an algorithm to implement ascending priority queue using singular linear linked list which has insert () function such that queue remains ordered list. Also implement remove() function. **(Mar. 10)**

Q. Write a function to implement insertion of an element in circular queue using link list. **(Dec. 10)**

Q. Write down C program showing various operations on a queue represented using a linked list.

(a) Insert five values in the queue.

(b) Print elements of the queue.

(c) Delete 2 elements from the queue.

(d) Print elements of the queue.

(e) Delete the remaining elements of the queue.

Q. Program for showing various operations on a queue represented using circular linked list.

4.11 Generalized Linked List

Q. Explain concept of generalized linked list with example.

Example 4.11.1 : Give representation of the generalized list.

$A = (1, 2, (3, (4, 5)), 6)$

Example 4.11.2 : Represent the following polynomial using GLL.

$X^{10}y^3z^2 + 2x^8y^3z^2 + 3x^8y^2z^2 + x^4y^4z + 6x^3y^4z + 2yz$

Example 4.11.3 : Represent the following polynomial using GLL.

$P(a,b,c) = a^{10}b^3c^2 + 6a^8b^3c^2 + 5a^8b^2c^2 + 2a^4b^4c + 2a^3b^4c + 8bc$

Example 4.11.4 : Represent the following polynomial using GLL.

$3x^{10}y^3z^2 + 5x^8y^3z^2 + 7x^8y^2z^2 + x^4y^4z + 6x^3y^4z + 9yz$

Example 4.11.5 : Represent the following polynomial using GLL.

$((x^{12} + 2x^9)y^4 + 4x^9y^2)z^3 + ((x^56x^3)y^5 + 3y)z$

Example 4.11.6 : Represent the following polynomial using GLL.

$3x^4y^3 + 5x^3y^3 + 7xy^3 + 3x^4y^6 + 5x^3y^6 + 7xy^6 + 6xy$

4.12 Dynamic Representation of Sparse Matrix

Q. Define sparse matrix. Briefly explain representation of sparse matrix with the help of list and array. **(May 13)**

Chapter 5 : Trees

Question Bank

(Answers of following Questions are available in the form of e-book)

5.1 Basic Terminology

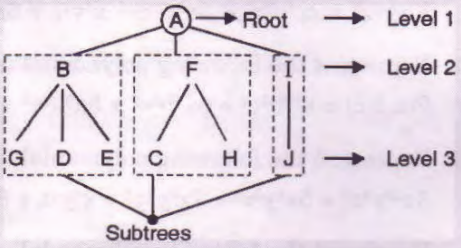
Q. Discuss with reference to trees – sibling. **(Dec. 09, Dec. 12, Dec.13, May 14)**

Q. Define terms - adjacent node and path. **(Dec. 12)**

Q. Define tree. **(Dec. 12, Dec. 13)**

Q. Discuss following with reference to trees – root, parent, child, path & degree of node.

Q. Discuss following with reference to trees – descendants, ancestors.

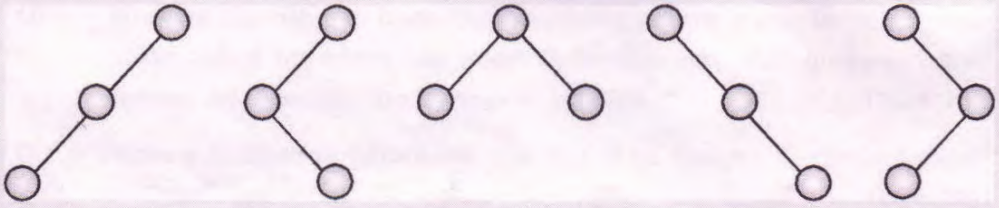


5.2 Binary Tree

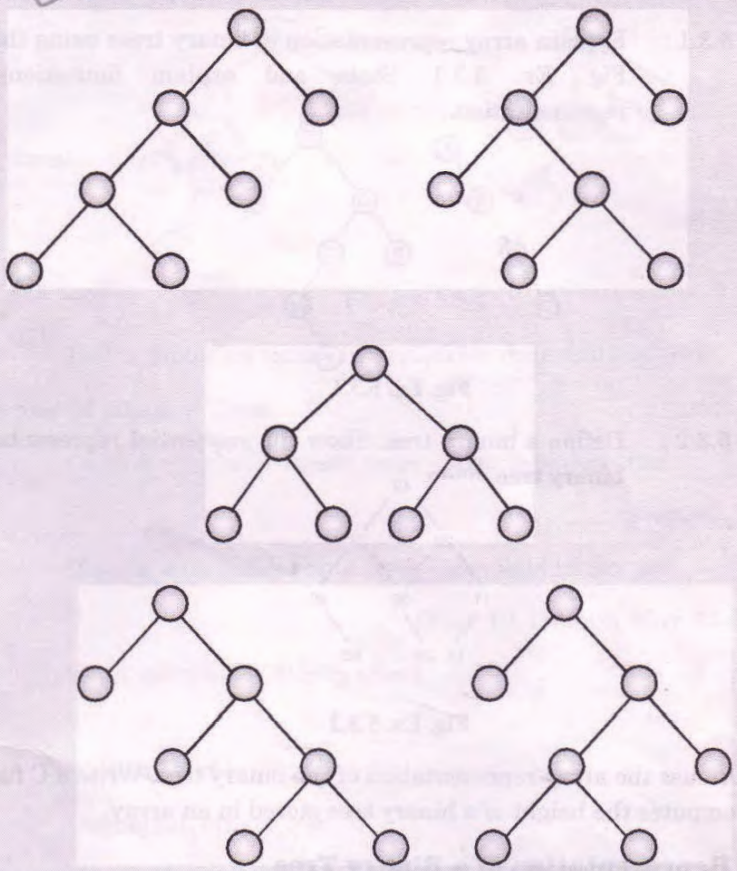
Q. Discuss with reference to trees - binary tree. **(Dec. 09, Mar 10, Dec. 12)**

Q. Answer the following : The height of a binary tree is the maximum number of edges in any root to leaf path. Define the maximum number of nodes in a binary tree of height h. **(May 13)**

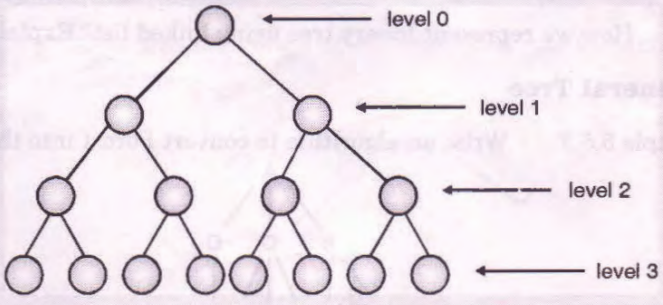
Q. Draw all possible binary trees with 3 nodes.



Q. Draw all possible binary trees T where T is a 2 tree with 4 external nodes.



Q. Show that maximum number of nodes in the binary tree of height h is $2^{h+1} - 1$.



Q. If I and E are the internal and external path lengths respectively of a binary tree with n Nodes, then determine the relationship between I and E.

5.3 Representation of a Binary Tree using an Array

Example 5.3.1 : Explain array representation of binary trees using the following Fig. Ex. 5.3.1. State and explain limitations of this representation.

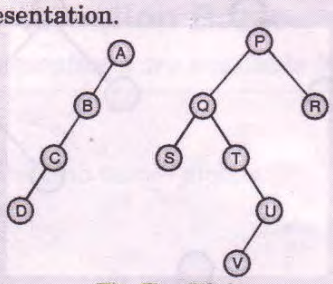


Fig. Ex. 5.3.1

Example 5.3.2 : Define a binary tree. Show the sequential representation of the binary tree given

Fig. Ex. 5.3.2

Q. Discuss the array representation of the binary tree. Write a C function that computes the height of a binary tree stored in an array.

5.4 Linked Representation of a Binary Tree

Q. Discuss with reference to trees - height of the tree

(Dec. 09, Mar 10, Dec. 10)

Q. How we represent binary tree using linked list? Explain with example.

5.5 A General Tree

Example 5.5.3 : Write an algorithm to convert Forest into the binary tree.

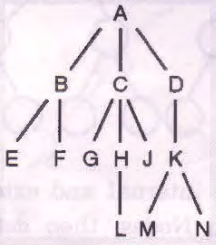


Fig. Ex. 5.5.3

Example 5.5.4 : Trace procedure to convert following forest into binary tree. **(Dec.12)**

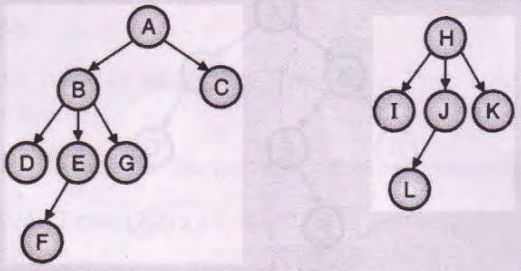


Fig. Ex. 5.5.4

Q. Define following term - Intermediate node and leaf node **(Dec. 12)**

5.6 Types of Binary Tree

Q. Discuss with reference to trees - strictly binary tree **(Dec. 09, Mar 10, May 13)**

Q. Discuss with reference to trees - complete binary tree **(Mar 10, Dec.10, May 11, May 12, May 14)**

Q. What is extended binary tree ? **(Dec.10)**

Q. Write a short note on weight balanced tree. **(May 12)**

Q. Define full binary tree **(May 14)**

5.7 Binary Tree Traversal

Q. Explain the preorder, Inorder and postorder traversal techniques of the binary tree with example. **(Dec. 09, Dec. 2011, Dec. 2013)**

Example 5.7.5 : Give traversal order of following tree into inorder, preorder and postorder. **(Dec. 10)**

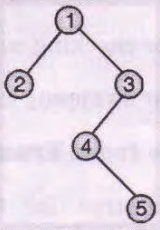


Fig. Ex. 5.7.5

Q. Write a non-recursive algorithm for preorder traversal of a binary tree. **(May 12)**

Q. Find a post order and preorder traversal of a following tree. **(Dec. 12)**

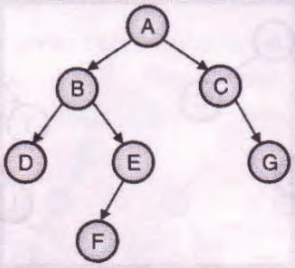


Fig. Q. 1

Q. Give the preorder and Inorder traversal of the tree given in Fig Q. 2.

(May 14)

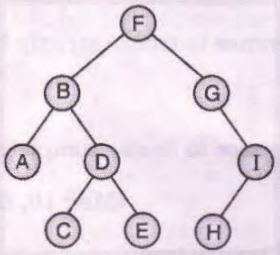


Fig. Q. 2

Q. Write down 'C' function for non-recursive inorder traversal

Q. Write down 'C' function for non-recursive preorder traversal.

Q. Write down 'C' function for non-recursive postorder traversal.

5.8 Basic Tree Operations

Q. Explain various operation possible with the trees with the suitable example.

Q. Write down recursive function for counting of leaf nodes in a tree.

Q. Write down function for checking equivalence of two binary trees.

Q. Write down function for swapping of left and right children of every node.

5.9 Creation of a Binary Tree from Traversal Sequence

Example 5.9.3 : Construct a tree for the given inorder and postorder to traversals.

Inorder : DGBAHEICF Postorder : GDBHIEFCA **(Mar 10)**

Q. Explain creation of binary tree from preorder and inorder traversals with example.

Q. Explain creation of Binary Tree from Postorder and Inorder Traversals with example.

Q. Construct a tree for the given inorder and postorder to traversals.

5.10 Binary Search Tree (BST)

Q. Discuss with reference to trees - binary search tree **(Mar 10, Dec. 11)**

Example 5.10.2 : Construct binary search tree for the following data :
10, 3, 15, 22, 6, 45, 65, 23, 78, 34, 5
Find its inorder, preorder and postorder traversal. **(Mar 10, May 13)**

Example 5.10.3 : Insert the following in a binary search tree.
(i) 7, 39, – 2, 0, 3, 42, 20, 5, 40
(ii) 15, 16, 5, 10, 8, 19, 4, 6, 17, 5, 21, 18, 10, 15, 6

Example 5.10.4 : In a binary search tree, given a key X, define successor (X) as the key which is the successor of X in the sorted order determined by an inorder tree walk. Define predecessor (X) similarly. Given X, write an algorithm to find its successor and predecessor.

Example 5.10.5 : A random binary search tree having n nodes, where $n = 2K - 1$, for some positive integer K, is to be organized into a perfectly balanced binary search tree. Outline an efficient algorithm for this task, The algorithm should not use any memory locations other than those already used by the random binary search tree.

Q. Write an C function to delete a node from a binary tree. **(May 11)**

Example 5.10.9 : Create a Binary Search Tree for the following data and do in-order, Preorder and Post-order traversal of the tree.
50, 60, 25, 40, 30, 70, 35, 10, 55, 65, 5. **(Dec. 11)**

Example 5.10.10 : Create a binary search tree for the following data :
50, 25, 75, 22, 40, 60, 80, 90, 15, 30 **(Dec.10)**

Example 5.10.11 : First insert 10 and then insert 24. After these insertions, delete 37 and then delete 22 from the following binary search tree. Draw the tree after each operation. **(May 12)**

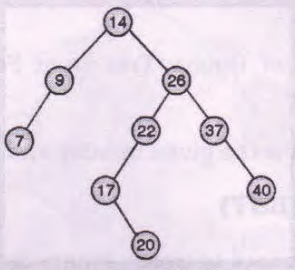


Fig. Ex. 5.10.11

Q. Create a Binary Search Tree for the following data and do Inorder, Preorder and Postorder traversal of the tree.40, 65,25, 55, 10,70,30,50,15,80,75 **(Dec.13)**

Example 5.10.12 : Given the following traversals create a binary tree from that. Also give the postorder traversal for the same.

Preorder = {7,10,4,3,1,2,8,11}

Inorder = {4,10,3,1,7,11,8,2} **(May 14)**

Q. Explain various operation possible with the binary search trees with the suitable example.

Program 5.10.1 : Write C program showing various operations on a binary search tree.

5.11 Threaded Binary Trees (TBT)

Example 5.11.2 :. Why is threaded binary tree required ? Draw a right in threaded binary tree for the given tree. **(Mar 10)**

Q. Write a short note on threaded binary tree. What are the advantages of threaded binary tree ? **(Dec.10, May 12, Dec. 12, Dec. 13)**

Q. Write a program in any language to create a threaded binary tree. **(May 11)**

Q. Explain the preorder, Inorder and postorder traversal techniques of the threaded binary tree with example.

Q. Explain the insert and delete operation related to threaded binary tree with example.

5.12 Application of Trees

Q. Write short note on applications of trees. **(Dec. 11)**

Example 5.12.3 : Obtain the expression tree from the following post fix representation ab+cde+**

Q. Write down program for creating an expression tree from a postfix expression and printing its preorder and inorder traversal sequence.

Q. How to convert an expression into binary tree? Explain with example.

5.13 AVL Trees

Example 5.13.1 : Draw diagram to show different stages during the building of AVL tree for the following sequence of keys : A, Z, B, Y, C, X, D, U, E. In each case show the balanced factor of all the nodes and name the type of rotation used for balancing.

Example 5.13.4 : Construct the AVL search tree by inserting the following elements in the order of their occurrence. **(Dec. 09)**

Q. Define an AVL tree. What is the meaning of height balanced tree ? How rebalancing is done in height balanced tree ? **(Dec. 10, May 11, May 13)**

Example 5.13.5 : Obtain an AVL tree by inserting one integer at a time in the following sequence. 150, 155, 160, 115, 110, 140, 120, 145, 130, 147, 170, 180. Show all the steps. **(May 11)**

Example 5.13.6 : Insert 1, 29, 32 and 13 in the following height balanced tree. For each insertion, draw the balanced tree using AVL rotation.

(May 12)

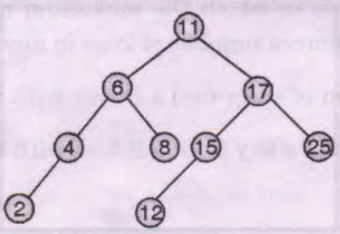


Fig. Ex. 5.13.6

Example 5.13.7 : Construct AVL tree for following data 10, 20, 30, 40, 50, 60, 70, 80 **(May 13)**

Example 5.13.9 : Construct a binary search tree for the following sequence. Also do the inorder and postorder traversal for the same

45,56,39,12,34,78,54,67,10,32,89,81 **(May 14)**

Q. Explain AVL tree with the help of an example also show insertion and deletion with the help of an example. **(May 14)**

Q. Explain the term balance factor with suitable example.

Q. Write down 'C' function for insertion of an element into an AVL Tree.

Q. Write down 'C' function for finding height of an AVL Tree.

Q. Write down C program to create an AVL tree and print the elements of the tree in

(a) Preorder

(b) Inorder and

(c) Postorder sequence.

5.14 B-Trees

Example 5.14.2 : . What are the advantages of multiway search tree in disc access ? Construct B-tree of order 5 for the following data :

1, 7, 6, 2, 11, 5, 10, 13, 12, 20, 16, 24, 3, 4, 18, 19, 14, 25 **(Mar 10)**

Example 5.14.3 : What are the advantages of multi way search tree over binary search tree ? Construct 2-3 tree for the following data

12, 50, 85, 6, 10, 37, 100, 120, 25, 70 **(May 13)**

Example 5.14.5 : Explain insertion operation in the 2-3 tree :

(i) if the parent has 2 children and

(ii) if the parent has 3 children. **(May 12)**

Q. Consider a B-tree in which the maximum number of keys in a node is 5. What is the minimum number of keys in any non-root node ? **(May 13)**

Q. Explain insertion of a key into a B-tree with example.

Q. Explain deletion of a key from a B-tree with example.

Chapter 6 : Graphs

Question Bank

(Answers of following Questions are available in the form of e-book)

6.1 Terminology and Representation

Q. Discuss following with reference to graphs.
- (i) Directed graph
- (ii) Undirected graph
- (iii) Degree of vertex
- (iv) Null graph. **(Dec. 09, Dec. 13)**

Q. What is graph ? **(Dec. 10, May 12, Dec.13)**

Q. What is spanning tree? **(Dec. 10, May 12, Dec.12, May 13, May 14)**

Q. Explain terms :
- (1) Path (2) Cycle
- (3) In-degree (4) Out-degree **(Dec. 11, May 12, Dec.13)**

Q. Define the term connected graph. **(May 14)**

Q. Discuss following with reference to graphs :
- (i) Incidence of graph
- (ii) Weighted graph
- (iii) Subgraph
- (iv) Component
- (v) Self edge or self loop
- (vi) Multigraph

Q. Write down applications of spanning tree.

6.2 Representation of Graphs

Q. Explain matrix and linked list representation of a graph.

(Dec. 09, Dec. 10, Dec. 11, Dec.13)

Example 6.2.1 : For the following graph obtain :
- (i) The in degree and out degree of each vertex,
- (ii) Its adjacency matrix
- (iii) Its adjacency list representation.

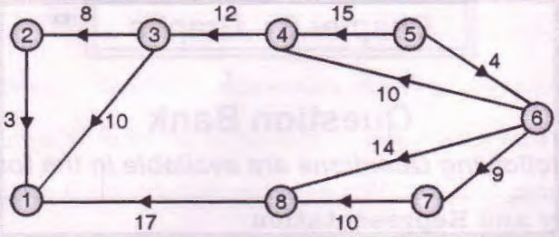


Fig. Ex. 6.2.1

Example 6.2.4 : Find the adjacency matrix for the graph shown below. **(Mar 10)**

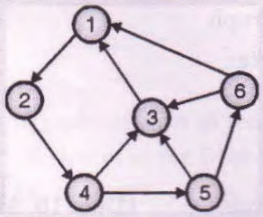


Fig. Ex. 6.2.4

6.3 Traversal of Graphs

Q. Which are the basic traversal techniques of the graph? Write the algorithm of any one of them. **(Dec. 09, Dec. 10, May 12, May 14)**

Example 6.3.3 : Draw the DFS spanning tree of the following graph. **(Dec. 12)**

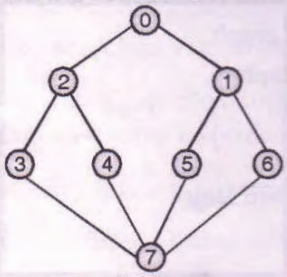


Fig. Ex. 6.3.3

Example 6.3.8 : Consider the graph shown below. Find depth-first and breadth first traversals of this graph starting at A. **(Mar 10)**

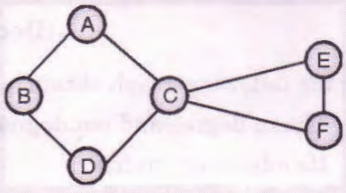


Fig. Ex. 6.3.8

Q. Explain BFS with example **(Dec. 10, May 12, May 14)**

Q. Compare the efficiencies of BFS and DFS. **(May 11, Dec. 11, Dec. 13)**

Q. Write an algorithm for breadth first search traversal of a graph. **(Dec. 11, Dec. 13)**

6.4 Connected Components

Q. Write Warshall algorithm for graph. **(Dec. 10)**

Example 6.4.2 : Obtain the adjacency matrix A for the following graph. Find A^2. Find outdegree of E and D nodes. **(May 12)**

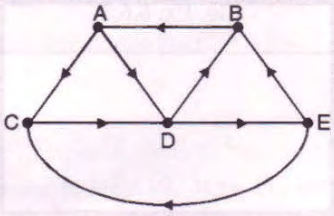


Fig. Ex. 6.4.2

6.5 Minimum Cost Spanning Tree

Example 6.5.2 : Find the minimum spanning tree of the following graph. **(Mar. 10)**

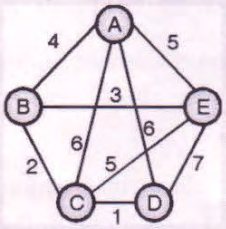


Fig. Ex. 6.5.2

Example 6.5.5 : Find minimum spanning tree for the graph shown in Fig. Ex. 6.5.5. **(May 13)**

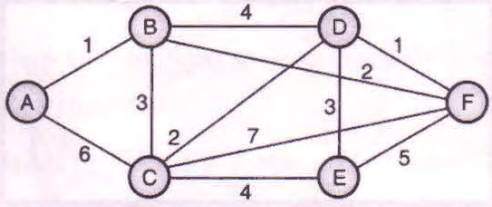


Fig. Ex. 6.5.5

Q. Compare and contrast Prim's and Kruskal's algorithm with the help of an example **(May 14)**

6.6 Shortest Path Algorithm

Example 6.6.3 : Apply Dijkstra's algorithm to find shortest path between vertex A and vertex F. **(Mar. 10)**

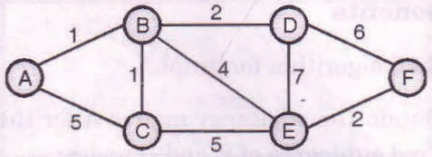


Fig. Ex. 6.6.3

Chapter 7 : Hashing and File Structures

Question Bank

(Answers of following Questions are available in the form of e-book)

7.1 Symbol Tables

Q. Write a short note on symbol table.

7.2 Representation of Symbol Table

Q. How we represent the symbol table?

7.3 Hash Tables

Q. What is hashing ? Explain various hashing functions. **(Dec.09, Dec.10, May 11, Dec.11, Dec.12, Dec.13, May 14)**

Q. What are the advantages of hashing ? Discuss problem of collision in hashing. Also discuss collision resolution techniques. **(Mar. 10)**

Example 7.3.1 : The integers given below are to be inserted in a hash table with 5 locations using chaining to resolve collisions. Construct hash table and use simplest hash function.

1, 2, 3, 4, 5, 10, 21, 22, 33, 34, 15, 32, 31, 48, 49, 50. **(Mar. 10)**

Q. What is hashing? Explain its resolution techniques. **(May 11, May 12, May 13)**

Example 7.3.2 : The keys 12, 18, 13, 2, 3, 23, 5 and 15 are inserted into an initially empty hash table of length 10 using open addressing with hash function h(k) = k mod 10 and linear probing. What is the resultant hash table ? **(May 13)**

Q. List the features of a good hash function **(May 14)**

Q. Write a short note on hash table data structure.

Q. Write a short note on linear probing and quadratic probing.

7.4 Introduction to File

Q. Explain the term : File, Field, Record, Database, Key. **(Dec. 09, May 11)**

7.5 File Organisation

Q. Explain sequential files. **(Dec. 09, Mar 10)**

Q. List various fundamental file organization techniques and explain each in brief. **(Dec. 10)**

Q. What is file structure? **(Dec.11)**

Q. List out primitive operations related with the sequential file organization.

7.6 Indexing and Hashing

Q. Define the term : direct files **(March 10, Dec.10)**

Q. What is the need of indexing in file organization?

7.7 Types of Indexes

Q. Explain indexed-sequential file. Also discuss the advantages and disadvantages of the same.

(Dec. 09, Mar. 10, May 11, Dec.11, May 12, Dec.12, May 13, Dec. 13, May 14)

Q. Define terms : Clustering index and secondary index.

Q. List out primitive operations related with the indexed sequential file organization.

7.8 Indexing and Hashing Comparisons

Q. Write a short note on indexing vs. hashing.

7.9 Multikey File Organization

Q. Explain various multiple key access file organization in brief.

(Dec. 10, Dec. 12, Dec. 13)

Q. Write a short note on comparison of various methods in the design of multikey file.

Chapter 8 : Sorting and Searching

Question Bank

(Answers of following Questions are available in the form of e-book)

8.1 Searching

Q. Define Searching.

8.2 Sequential Search

Q. Explain linear search with suitable example.

Q. Write down 'C' function for sequential search in a sorted array.

8.3 Binary Search

Q. Write recursive C function for binary search with suitable example and analyze its time complexity for best and worst cases.

Q. Explain binary search in brief.

Q. Given an array size in containing integers sorted in descending order, write 'C' functions for the following :

(1) To search a given number in array using recursion.

(2) To search a given number in array without using recursion.

Q. Compare linear search and binary Search.

8.4 Sorting

Q. Explain the terms : Sort stability and sort efficiency.

Q. Explain internal sorting, external sorting and passes.

8.5 Insertion Sort

Q. Write 'C' function to sort an array of size containing integers using insertion sort. Obtain number of push downs required and time complexity of your program in best and worst case. When insertion sort technique is preferred ?

8.6 Bubble Sort

Q. Write down algorithm for bubble sort, explain it with example and write time complexity.

Q. Write a C program for sorting 100 integer numbers by bubble sort. Discuss the worst case time complexity of an algorithm.

8.7 Selection Sort

Q. What is basic operation performed in selection sort technique ?

Q. Write pseudo 'C' code to accept 'N' numbers at random.

Q. Compare the three sorting algorithms :

(a) Bubble sort

(b) Selection sort

(c) Insertion sort

8.8 Quick Sort

Q. Write down algorithm for quick sort, explain it with example and write time complexity.

Q. Explain role of pivot in efficiency of quick sort.

8.9 Two-Way Merge Sort

Q. Write down algorithm for merge sort, explain it with example and write time complexity.

❑❑❑