**Question Bank**

**Unit – 1**

**Objective Questions**

**Q.1** The complexity of multiplying two matrices of order m\*n and n\*p is

**(A)** mnp **(B)** mp

**(C)** mn **(D)** np

**Q.2** Merging 4 sorted files containing 50, 10, 25 and 16 records will take\_\_\_\_time

**(A)** O (100) **(B)** O (200)

**(C)** O (175) **(D)** O (125)

**Q.3** Consider a linked list of n elements. What is the time taken to insert an element after an element pointed by some pointer?

**(A)** O (1) **(B)** O log2 n

**(C)** O (n) **(D)** O n log2 n

**Q.4** The smallest element of an array’s index is called its

**(A)** lower bound. **(B)** upper bound.

**(C)** range. **(D)** extraction.

**Q.5** In a circular linked list

**(A)** components are all linked together in some sequential manner.

**(B)** there is no beginning and no end.

**(C)** components are arranged hierarchically.

**(D)** forward and backward traversal within the list is permitted.

**Q.6** The minimum number of multiplications and additions required to evaluate the polynomial

P = 4x3+3x2-16x+45 is

**(A)** 6 & 3 **(B)** 4 & 2

**(C)** 3 & 3 **(D)** 8 & 3

**Q.7** In a linked list with n nodes, the time taken to insert an element after an element pointed by

some pointer is

**(A)** 0 (1) **(B)** 0 (log n)

**(C)** 0 (n) **(D)** 0 (n 1og n)

**Q.8** What data structure would you mostly likely see in a nonrecursive implementation of a recursive algorithm?

**(A)** Stack **(B)** Linked list

**(C)** Queue **(D)** Trees

**Q.9** Let the following circular queue can accommodate maximum six elements with the

following data

front = 2 rear = 4

queue = \_\_\_\_\_\_\_; L, M, N, \_\_\_, \_\_\_

What will happen after ADD O operation takes place?

**(A)** front = 2 rear = 5

queue = \_\_\_\_\_\_; L, M, N, O, \_\_\_

**(B)** front = 3 rear = 5

queue = L, M, N, O, \_\_\_

**(C)** front = 3 rear = 4

queue = \_\_\_\_\_\_; L, M, N, O, \_\_\_

**(D)** front = 2 rear = 4

queue = L, M, N, O, \_\_\_

**Q.10** A linear collection of data elements where the linear node is given by means of pointer is

called

**(A)** linked list **(B)** node list

**(C)** primitive list **(D)** None of these

**Q.11** Representation of data structure in memory is known as:

**(A)** recursive **(B)** abstract data type

**(C)** storage structure **(D)** file structure

**Q.12** If the address of A[1][1] and A[2][1] are 1000 and 1010 respectively and each element occupies 2 bytes then the array has been stored in \_\_\_\_\_\_\_\_\_ order.

**(A)** row major **(B)** column major

**(C)** matix major **(D)** none of these

**Q.13** An adjacency matrix representation of a graph cannot contain information of :

**(A)** nodes **(B)** edges

**(C)** direction of edges **(D)** parallel edges

**Q.14** Time complexities of three algorithms are given. Which should execute the slowest for large values of N?

**(A)** 1 2 *O N* **(B)** *O**N*

**(C)** *O*log *N***(D)** *None of these*

**Q.16** How does an array differ from an ordinary variable?

**Q.16** Which of the following operations is performed more efficiently by doubly linked list than by singly linked list?

**(A)** Deleting a node whose location in given

**(B)** Searching of an unsorted list for a given item

**(C)** Inverting a node after the node with given location

**(D)** Traversing a list to process each node

**Q.17** The extra key inserted at the end of the array is called a,

**(A)** End key. **(B)** Stop key.

**(C)** Sentinel. **(D)** Transposition.

**Q.18** The time required to delete a node x from a doubly linked list having n nodes is

**(A)** O (n) **(B)** O (log n)

**(C)** O (1) **(D)** O (n log n)

**Part A**

**Q.1** Which sorting algorithm is easily adaptable to singly linked lists? Explain

your answer. **(4)**

**Q 2.** Determine the frequency counts for all statements in the following program

segment.

for (i=1; i <= n; i ++)

for (j = 1; j <= i; j++)

for (k =1; k <= j; k++)

y ++;

**Q 3 .** Write an algorithm to count number of nodes in the circular linked list. **(3)**

**Q 4.** Write an algorithm to insert a node in between any two nodes in a linked list **(4)**

**Q.5** What is the difference between a grounded header link list and a circular header

link list? **(3)**

**Q 6.** A linear array A is given with lower bound as 1. If address of A[25] is 375 and

A[30] is 390, then find address of A[16]. **(4)**

**Q7.** Write an algorithm to insert a node p at the end of a linked list. **(5)**

**Q8.** Write an algorithm that counts number of nodes in a linked list. **(5)**

**Q9.** Write an algorithm to add an element at the end of circular linked list. **(5)**

**Q10.** Delete a given node from a doubly linked list. **(4)**

**Part B**

**Q.1** Explain an efficient way of storing a sparse matrix in memory. Write a

module to find the transpose of a sparse matrix stored in this way. **(10)**

**Q.2** Two linked lists contain information of the same type in ascending order.

Write a module to merge them to a single linked list that is sorted. **(7)**

**Q.3** An, array, A contains n unique integers from the range x to y (x and y

inclusive where n=y-x). That is, there is one member that is not in A. Design

an O(n) time algorithm for finding that number. **(8)**

**Q.4** Bubble sort algorithm is inefficient because it continues execution even after

an array is sorted by performing unnecessary comparisons. Therefore, the

number of comparisons in the best and worst cases are the same. Modify the

algorithm in such a fashion that it will not make the next pass when the array

is already sorted. **(12)**

**Q.5** What do you mean by complexity of an algorithm? Explain the meaning of

worst case analysis and best case analysis with an example. **(8)**

**Q.6** Explain the method to calculate the address of an element in an array. A

25\*4 matrix array DATA is stored in memory in ‘row-major order’. If base

address is 200 and 4 words per memory cell. Calculate the address of

DATA [12, 3] . **(8)**

**Q.7** Write an algorithm to insert a node in the beginning of the linked list. **(7)**

**Q.8** Why do we use asymptotic notation in the study of algorithm? Describe

commonly used asymptotic notations and give their significance. **(8)**

**Q.9** What is a linear array? Explain how two dimensional arrays are represented in

memory. **(8)**

**Q.10** Write a complete programme in C to create a single linked list. Write

functions to do the following operations

(i) Insert a new node at the end

(ii) Delete the first node **(8)**

**Q.11** Define a sparse matrices. Explain the representation of a 4X4 matrix using

linked list. **(8)**

**Q.12** Write a procedure to reverse a singly linked list. **(8)**

**Q 13.** Define a sparse matrix. Explain different types of sparse matrices? Show how a

triangular array is stored in memory. Evaluate the method to calculate address of

any element ajk of a matrix stored in memory. **(10)**

**Q 14.** Show the linked representation of the following two polynomials.



Build a procedure for adding two polynomials stored in linked lists. Verify

steps of your procedure for the above two polynomials. **(7)**

**Q 16.** What is a sparse matrix? How is it stored in the memory of a computer? Write a

function to find the transpose of a sparse matrix using this representation. **(8)**

**Q 16.** Write an algorithm for finding solution to the Towers of Hanoi problem. Explain

the working of your algorithm (with 4 disks) with diagrams. **(7)**

**Q 17.** Suppose we have divided n elements in to m sorted lists. Explain how to

produce a single sorted list of all n elements in time O (n log m )? **(7)**

**Q.18** Define the term array. How are two-dimensional arrays represented in

memory? Explain how address of an element is calculated in a two

dimensional array. **(8)**

**Q.19** What is an algorithm? What are the characteristics of a good algorithm?

**Q.20** How do you find the complexity of an algorithm? What is the relation

between the time and space complexities of an algorithm? Justify your answer with an example.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

**Unit 2**

**Objective Questions**

**Q.1** The postfix form of the expression ABCD−EF / G is

**(A)** ABCDE −FG /**(B)** AB CDE −F G /

**(C)** AB CDE −F G / **(D)** AB CDE −F G /

**Q.2** A linear list of elements in which deletion can be done from one end (front) and insertion

can take place only at the other end (rear) is known as a

**(A)** queue. **(B)** stack.

**(C)** tree. **(D)** linked list.

**Q.3** What is the postfix form of the following prefix expression -A/B\*C$DE

**(A)** ABCDE$\*/- **(B)** A-BCDE$\*/-

**(C)** ABC$ED\*/- **(D)** A-BCDE$\*/

**Q.4** The data structure required to evaluate a postfix expression is

**(A)** queue **(B)** stack

**(C)** array **(D)** linked-list

**Q.5** The data structure required to check whether an expression contains balanced parenthesis is

**(A)** Stack **(B)** Queue

**(C)** Tree **(D)** Array

**Q.6** The postfix form of A\*B+C/D is

**(A)** \*AB/CD+ **(B)** AB\*CD/+

**(C)** A\*BC+/D **(D)** ABCD+/\*

**Q.7** What is the postfix form of the following prefix *\*+ab–cd*

**(A)** *ab+cd–\** **(B)** *abc+\*–*

**(C)** *ab+\*cd–* **(D)** *ab+\*cd–*

**Q.8** A stack is to be implemented using an array. The associated declarations are:

int stack [100];

int top = 0;

Give the statement to perform push operation.

**Q.9** Assume that a queue is available for pushing and popping elements. Given an input sequence a, b, c, (c be the first element), give the output sequence of elements if the rightmost element given above is the first to be popped from the queue.

**Q.10** A queue is a,

**(A)** FIFO (First In First Out) list. **(B)** LIFO (Last In First Out) list.

**(C)** Ordered array. **(D)** Linear tree.

**Q.11** Which data structure is needed to convert infix notation to postfix notation?

**(A)** Branch **(B)** Queue

**(C)** Tree **(D)** Stack

**Q.12** The prefix form of A-B/ (C \* D ^ E) is,

**(A)** -/\*^ACBDE **(B)** -ABCD\*^DE

**(C)** -A/B\*C^DE **(D)** -A/BC\*^DE

**Q.13** What is the result of the following operation

Top (Push (S, X))

**(A)** X **(B)** null

**(C)** S **(D)** None of these.

**Q.14** The prefix form of an infix expression p q −r \* t is

**(A)** pq −\*rt . **(B)** −pqr \* t .

**(C)** −pq \* rt . **(D)** −\* pqrt .

**Q.16** Which data structure is used for implementing recursion?

**(A)** Queue. **(B)** Stack.

**(C)** Arrays. **(D)** List.

**Q.16** The equivalent prefix expression for the following infix expression (A+B)-(C+D\*E)/F\*G is

**(A)** -+AB\*/+C\*DEFG **(B)** /-+AB\*+C\*DEFG

**(C)** -/+AB\*+CDE\*FG **(D)** -+AB\*/+CDE\*FG

**Q.17** The equivalent prefix expression for the following infix expression (A+B)-(C+D\*E)/F\*G is

**(A)** -+AB\*/+C\*DEFG **(B)** /-+AB\*+C\*DEFG

**(C)** -/+AB\*+CDE\*FG **(D)** -+AB\*/+CDE\*FG

**Part A**

**Q1.** Write down any four application of a stack. **(4)**

**Q2.** Convert the following infix expression into a postfix expression (Show steps)

(i)ABD/ E −FG H/ k**(4)**

(ii) A B D/E −FG (4)

(iii) a bc d/e f g .

**Q.3** What are stacks? How can stacks be used to check whether an expression is

correctly parenthized or not. For eg(()) is well formed but (() or )()( is not.**(7)**

**Q4.** Convert the following Infix expression to Postfix form using a stack:

*x* + *y* \* *z* + (*p* \* *q* + *r* ) \* *s*, Follow usual precedence rule and assume that the

expression is legal. **(7)**

**Q5.** Define a stack. Describe ways to implement stack. **(5)**

**Q6.** Can a Queue be represented by circular linked list with only one pointer

pointing to the tail of the queue? Substantiate your answer using an example.**(5)**

**Q7.** Convert the following infix expressions to postfix notation

(i) A+((B+C)\*(D+E)+F/G)

(ii) A B CD **(4)**

**Q8.** Suggest a way of implementing two stacks in one array such that as long as

space is there in an array, you should be able to add an element in either stack.

Using proposed method, write algorithms for push and pop operations for both

the stacks. **(6)**

**Q9.** Write down any four applications of queues. **(4)**

**Part B**

**Q.1** Reverse the order of elements on a stack S

(i) using two additional stacks.

(ii) using one additional queue. **(9)**

**Q.2** Write an algorithm to evaluate a postfix expression. Execute your algorithm

using the following postfix expression as your input : a b + c d +\*f . **(7)**

**Q.3** What are circular queues? Write down routines for inserting and deleting

elements from a circular queue implemented using arrays. **(7)**

**Q.4** Implement a Queue using a singly linked list L. The operations INSERT and

DELETE should still take O (1) time. **(6)**

**Q.5** Explain how to implement two stacks in one array A[1..n] in such a way that

neither stack overflows unless the total number of elements in both stacks

together is n. The PUSH and POP operations should run in O(1) time. **(10)**

**Q.6** Let P be a pointer to a singly linked list. Show how this list may be used as a

stack. That is, write algorithms to push and pop elements. Specify the value of P

when the stack is empty. **(8)**

**Q.7** Execute your algorithm to convert an infix expression to a post fix expression

with the following infix expression on your input

mn\*k p/g / ba b / c**(8)**

**Q.8** A double ended queue is a linear list where additions and deletions can be

performed at either end. Represent a double ended queue using an array to store

elements and write modules for additions and deletions. **(8)**

**Q.9** Devise a representation for a list where insertions and deletions can be made at

either end. Such a structure is called Deque (Double ended queue). Write

functions for inserting and deleting at either end. **(8)**

**Q.10** Execute your algorithm to convert an infix expression to a post fix expression

with the following infix expression as input

Q A B/C DE / FG H/ I **(8)**

**Q11.** Using array to implement the queue structure, write an algorithm/program to

(i) Insert an element in the queue.

(ii) Delete an element from the queue. **(9)**

**Q12.** Write an algorithm to evaluate an expression given in postfix notation. Show the

execution of your algorithm for the following expression.

AB^CD-EF/GH+/+\* **(7)**

**Q13.** Write an algorithm to convert an infix expression into postfix expression. **(8)**

**Q14.** Using stacks, write an algorithm to determine whether the infix expression has

balanced parenthesis or not. **(7)**

**Q16.** Implement a stack using linked list. Show both the PUSH and POP operations.

**Unit 3**

Q1 Let A be an adjacency matrix of a graph G. The th ij entry in the matrix K A , gives

**(A)** The number of paths of length K from vertex Vi to vertex Vj.

**(B)** Shortest path of K edges from vertex Vi to vertex Vj.

**(C)** Length of a Eulerian path from vertex Vi to vertex Vj.

**(D)** Length of a Hamiltonian cycle from vertex Vi to vertex Vj.

**Q.2** If a node having two children is deleted from a binary tree, it is replaced by its

**(A)** Inorder predecessor **(B)** Inorder successor

**(C)** Preorder predecessor **(D)** None of the above

**Q.3** For an undirected graph with n vertices and e edges, the sum of the degree of each vertex is equal to

**(A)** 2n **(B)** (2n-1)/2

**(C)** 2e **(D)** e2/2

**Q.4** A full binary tree with 2n+1 nodes contain

**(A)** n leaf nodes **(B)** n non-leaf nodes

**(C)** n-1 leaf nodes **(D)** n-1 non-leaf nodes

**Q.5** A full binary tree with n leaves contains

**(A)** n nodes. **(B)** log n 2 nodes.

**(C)** 2n –1 nodes. **(D)** n 2 nodes.

**Q.6** An undirected graph G with n vertices and e edges is represented by adjacency list. What is the time required to generate all the connected components?

**(A)** O (n) **(B)** O (e)

**(C)** O (e+n) **(D)** O 2 e

**Q.7** A graph with n vertices will definitely have a parallel edge or self loop of the total number of edges are

**(A)** more than n **(B)** more than n+1

**(C)** more than (n+1)/2 **(D)** more than n(n-1)/2

**Q.8** The maximum degree of any vertex in a simple graph with *n* vertices is

**(A)** *n–1* **(B)** *n+1*

**(C)** *2n–1* **(D)** *n*

**Q.9** The data structure required for Breadth First Traversal on a graph is

**(A)** queue **(B)** stack

**(C)** array **(D)** tree

**Q.10** The number of different directed trees with 3 nodes are

**(A)** 2 **(B)** 3

**(C)** 4 **(D)** 5

**Q.11** One can convert a binary tree into its mirror image by traversing it in

**(A)** inorder **(B)** preorder

**(C)** postorder **(D)** any order

**Q.12** One can convert a binary tree into its mirror image by traversing it in

**(A)** inorder **(B)** preorder

**(C)** postorder **(D)** any order

**Q.13** The number of leaf nodes in a complete binary tree of depth d is

**(A)** 2d **(B)** 2d–1+1

**(C)** 2d+1+1 **(D)** 2d+1

**Q.14** The pre-order and post order traversal of a Binary Tree generates the same output. The tree can have maximum

**(A)** Three nodes **(B)** Two nodes

**(C)** One node **(D)** Any number of nodes

**Q.16** A graph with n vertices will definitely have a parallel edge or self loop if the total number of edges are

**(A)** greater than n–1 **(B)** less than n(n–1)

**(C)** greater than n(n–1)/2 **(D)** less than n2/2

**Q.16** A binary tree of depth “d” is an almost complete binary tree if

**(A)** Each leaf in the tree is either at level “d” or at level “d–1”

**(B)** For any node “n” in the tree with a right descendent at level “d” all the left

descendents of “n” that are leaves, are also at level “d”

**(C)** Both **(A)** & **(B)**

**(D)** None of the above

**Q.17** In Breadth First Search of Graph, which of the following data structure is used?

**(A)** Stack. **(B)** Queue.

**(C)** Linked List. **(D)** None of the above.

**Q.18** For an undirected graph G with n vertices and e edges, the sum of the degrees of each vertex is

**(A)** ne **(B)** 2n

**(C)** 2e **(D)** en

**Part A**

**Q.1** What are expression trees? Represent the following expression using a tree.

Comment on the result that you get when this tree is traversed in Preorder,

Inorder and postorder. (a-b) / ((c\*d)+e) **(6)**

**Q.2** Taking a suitable example explains how a general tree can be represented as a

Binary Tree. **(6)**

**Q.3** What are the different ways of representing a graph? Represent the following

graph using those ways. **(6)**

****

**Q.4** Give the adjacency matrix for the following graph: **(4)**

**Q.5** Create a heap with following list of keys:

8, 20, 9, 4, 16, 10, 7, 22, 3, 12 **(5)**

**Q.6** Construct a complete binary tree with depth 3 for this tree which is maintained

in memory using linked representation. Make the adjacency list and adjacency matrix for this tree. (6)

**Q7.** A Binary tree has 9 nodes. The inorder and preorder traversals of the tree

yields the following sequence of nodes:

Inorder : E A C K F H D B G

Preorder: F A E K C D H G B

Draw the tree. Explain your algorithm. **(7)**

**Q.8** How will you represent a max-heap sequentially? Explain with an example. **(4)**

**Q9.** Construct the binary tree for the following sequence of nodes in preorder and

inorder respectively.

Preorder : G, B, Q, A, C, K, F, P, D, E, R, H

Inorder: Q, B, K, C, F, A, G, P, E, D, H, R **(4)**

**Q10.** Give the algorithm to construct a binary tree where the yields of preorder and

post order traversal are given. **(6)**

**Q11.** Draw a picture of the directed graph specified below:

G = ( V, E)

V(G) = {1, 2, 3, 4, 5, 6}

E(G) = {(1,2), (2, 3), (3, 4), (5,1), (5, 6), (2, 6), (1, 6), (4, 6), (2, 4)}

Obtain the following for the above graph:

(i) Adjacency matrix.

(ii) React ability matrix. **(7)**

**Q.12** Draw a binary tree from its inorder and preorder traversal sequences given as

follows:

Inorder : d b g e h a c n f

Preorder : a b d e g h c f n **(7)**

**Part B**

**Q.1** Draw the expression tree of the following infix expression. Convert it in to

Prefix and Postfix expressions.

a bc \* d ef \* g h**(9**

**Q.2** Given a set of input representing the nodes of a binary tree, write a non

recursive algorithm that must be able to output the three traversal orders.

**Q.3** How do you rotate a Binary Tree? Explain right and left rotations with the help of

an example. **(8)**

**Q.4** Show the result of running BFS and DFS on the directed graph given below

using vertex 3 as source. Show the status of the data structure used at each

stage. **(8)**

****

**Q.5** Explain the representations of graph. Represent the given graph using any two

methods **(8)**

****

**Q.6** Two Binary Trees are similar if they are both empty or if they are both nonempty

and left and right sub trees are similar. Write an algorithm to determine

if two Binary Trees are similar. **(8)**

**Q.7** The degree of a node is the number of children it has. Show that in any binary tree, the

number of leaves are one more than the number of nodes of degree 2 **(8)**

**Q.8** Write the non-recursive algorithm to traverse a tree in preorder. **(8)**

**Q.9** Draw the complete undirected graphs on one, two, three, four and five

vertices. Prove that the number of edges in an n vertex complete graph is

n(n-1)/2. **(8)**

****

**Q.10** Write an algorithm which does depth first search through an un-weighted

connected graph. In an un-weighted graph, would breadth first search or depth

first search or neither find a shortest path tree from some node? Why? **(8)**

**Q.11** Write a non recursive algorithm to traverse a binary tree in inorder. **(8)**

**Q.12** Which are the two standard ways of traversing a graph? Explain them with an

example of each. **(8)**

**Q.13** Consider the following specification of a graph G

VG1,2,3,4

EG1,2, 1,3, 3,3, 3,4, 4,1

(i) Draw an undirected graph.

(ii) Draw its adjacency matrix.

**Q.14** Write an algorithm to insert an element to a max-heap that is represented

sequentially. **(8)**

**Q.16** Construct a binary tree whose nodes in inorder and preorder are given as

follows:

Inorder : 10, 16, 17, 18, 20, 25, 30, 35, 38, 40, 50

Preorder: 20, 16, 10, 18, 17, 30, 25, 40, 35, 38, 50 **(10)**

**Q.16** Given the following inorder and preorder traversal reconstruct a binary tree

Inorder sequence D, G, B, H, E, A, F, I, C

Preorder sequence A, B, D, G, E, H, C, F, I **(8)**

**Q.17** What is a Binary Tree? What is the maximum number of nodes possible in a

Binary Tree of depth d. Explain the following terms with respect to Binary

trees

(i) Strictly Binary Tree (ii) Complete Binary Tree (iii) Almost

Complete Binary Tree **(8)**

**Q.18** Show the result of running BFS and DFS on a directed graph given below

using vertex 1 as source. Show the status of the data structure used at each

stage. **(10)**

****

**Q.19** Define adjacency matrix and make the same for the following undirected

graph. **(8)**

****

**Q.20** Show the linked representation of the above graph. **(8)**

**Q.21** What do you understand by tree traversal? Write a procedure for traversing a

binary tree in preorder and execute it on the following tree. **(8)**

**Q22.** Sort the following list using Heap Sort technique, displaying each step.

20, 12, 25 6, 10, 16, 13 **(7)**

**Q.23.** Give the adjacency matrix and adjacency list of the following graphs.

****

**Q24.** Sort the following list using Heap Sort

66, 33, 40, 20, 50, 88, 60, 11, 77, 30, 45, 65. **(7)**

**Q25**. What are the two phases in heap sort algorithm? Sort the following data

using heap sort and show all the intermediate steps.

88, 12, 91, 23, 10, 36, 45, 55, 16, 39, 81

**Q.29** Draw the complete undirected graphs on one, two, three, four and five

vertices. Prove that the number of edges in an n vertex complete graph is

n(n-1)/2. **(8)**

**Unit 4**

**Q.1** If h is any hashing function and is used to hash n keys in to a table of size m, where n<=m, the expected number of collisions involving a particular key x is :

**(A)** less than 1. **(B)** less than n.

**(C)** less than m. **(D)** less than n/2.

**Q.2** A technique for direct search is

**(A)** Binary Search **(B)** Linear Search

**(C)** Tree Search **(D)** Hashing

**Q.3** You have to sort a list L consisting of a sorted list followed by a few “random” elements. Which of the following sorting methods would be especially suitable for such a task?

**(A)** Bubble sort **(B)** Selection sort

**(C)** Quick sort **(D)** Insertion sort

**Q.4** The searching technique that takes O (1) time to find a data is

**(A)** Linear Search **(B)** Binary Search

**(C)** Hashing **(D)** Tree Search

**Q.5** In worst case Quick Sort has order

**(A)** O (n log n) **(B)** O (n2/2)

**(C)** O (log n) **(D)** O (n2/4)

**Q.6** A sort which relatively passes through a list to exchange the first element with any element less than it and then repeats with a new first element is called

**(A)** insertion sort. **(B)** selection sort.

**(C)** heap sort. **(D)** quick sort.

**Q.7** Which of the following sorting algorithms does not have a worst case running time of 2 O n ?

**(A)** Insertion sort **(B)** Merge sort

**(C)** Quick sort **(D)** Bubble sort

**Q.8** The quick sort algorithm exploit \_\_\_\_\_\_\_\_\_ design technique

**(A)** Greedy **(B)** Dynamic programming

**(C)** Divide and Conquer **(D)** Backtracking

**Q.9** The complexity of searching an element from a set of n elements using Binary search algorithm is

**(A)** O(n) **(B)** O(log n)

**(C)** O(n2) **(D)** O(n log n)

**Q.10** Which of the following sorting methods would be most suitable for sorting a list which is almost sorted

**(A)** Bubble Sort **(B)** Insertion Sort

**(C)** Selection Sort **(D)** Quick Sort

**Q.11** Quick sort is also known as

**(A)** merge sort **(B)** heap sort

**(C)** bubble sort **(D)** none of these

**Q.12** The goal of hashing is to produce a search that takes

**(A)** *O(1)* time **(B)** *O(n2 )* time

**(C)** *O(log n )* time **(D)** *O(n log n )* time

**Q.13** The best average behaviour is shown by

**(A)** Quick Sort **(B)** Merge Sort

**(C)** Insertion Sort **(D)** Heap Sort

**Q.14** Which sorting algorithm is best if the list is already sorted? Why?

**Q.16** What is the time complexity of Merge sort and Heap sort algorithms?

**Q.16** Consider that n elements are to be sorted. What is the worst case time complexity of Bubblesort?

**(A)** O(1) **(B)** O(log2n)

**(C)** O(n) **(D)** O(n2)

**Q.17** A characteristic of the data that binary search uses but the linear search ignores is

the\_\_\_\_\_\_\_\_\_\_\_.

**(A)** Order of the elements of the list.

**(B)** Length of the list.

**(C)** Maximum value in list.

**(D)** Type of elements of the list.

**Q.18** The worst case of quick sort has order

**(A)** O(n2) **(B)** O(n)

**(C)** O (n log2 n) **(D) O** (log2 n)

**Part A**

**Q.1** How many key comparisons and assignments an insertion sort makes in its

worst case?

**Q.2** What is the best case complexity of quick sort and outline why it is so. How

could its worst case behaviour arise?

**Q3.** Write an algorithm to sort a given list using Quick sort method. Describe the

behaviour of Quick sort when input is already sorted.

**Part B**

**Q.1** What is quick sort? Sort the following array using quick sort method.

24 56 47 35 10 90 82 31.

**Q.2** Sort the following sequence of keys using merge sort.

66, 77, 11, 88, 99, 22, 33, 44, 55.

**Q.3** The following values are to be stored in a hash table

25, 42, 96, 101, 102, 162, 197

Describe how the values are hashed by using division method of hashing with

a table size of 7. Use chaining as the method of collision resolution.

**Q.4** Describe insertion sort with a proper algorithm. What is the complexity of

insertion sort in the worst case?

**Q.5** What do you mean by hashing? Explain any five popular hash functions.

**Q.6** Write an algorithm to merge two sorted arrays into a third array. Do not sort

the third array.

**Q.7** Define Hashing. How do collisions happen during hashing? Explain the

different techniques resolving of collision.

**Q.8** What do you mean by hash clash? Explain in detail any one method to resolve

hash collisions.

**Q9.** Execute quick algorithm on the following data till two key values are placed in

their position 12,34,45,16,4,11,7,8,5,14,35,89,43,21.

**Q 10** Sort the following array of elements using quick sort *{3 1 4 1 5 9 2*

*6 5 3 5 8}*

**Q.11** Execute your algorithm for two passes using the following list as input:

66, 33, 40, 20, 50, 88, 60, 11, 77, 30, 45, 65

Describe the behaviour of Quick sort when the input is already sorted.

**Q12.** Write down the algorithm of quick sort.

**Q13.** Draw the 11 item hash table resulting from hashing the keys: 12, 44, 13, 88,

23, 94, 11, 39, 20, 16 and 5 using the hash function h(i) = (2i+5) mod 11.

**Q14.** Write an algorithm for selection sort. Describe the behaviours of selection sort

when the input is already sorted.

**Q16.** Explain Hash Tables, Hash function and Hashing Techniques?

**Q16.** Define hashing. Describe any two commonly used hash functions. Describe one method of collision resolution.

**Q.17** Compare and contrast various sorting techniques with respect to memory

space and computing time.

**Unit 5**

**Q.1** B Trees are generally

**(A)** very deep and narrow **(B)** very wide and shallow

**(C)** very deep and very wide **(D)** cannot say

**Q.2** If a node in a BST has two children, then its inorder predecessor has

**(A)** no left child **(B)** no right child

**(C)** two children **(D)** no child

**Q.3** A binary tree in which if all its levels except possibly the last, have the maximum number of nodes and all the nodes at the last level appear as far left as possible, is known as

**(A)** full binary tree. **(B)** AVL tree.

**(C)** threaded tree. **(D)** complete binary tree.

**Q.4** A B-tree of minimum degree t can maximum \_\_\_\_\_ pointers in a node.

**(A)** t–1 **(B)** 2t–1

**(C)** 2t **(D)** t

**Q.5** A BST is traversed in the following order recursively: Right, root, left

The output sequence will be in

**(A)** Ascending order **(B)** Descending order

**(C)** Bitomic sequence **(D)** No specific order

**Q.6** One of the major drawback of B-Tree is the difficulty of traversing the keys sequentially.

**Q.8** In order to get the information stored in a Binary Search Tree in the descending order, one should traverse it in which of the following order?

**(A)** left, root, right **(B)** root, left, right

**(C)** right, root, left **(D)** right, left, root

**Part A**

**Q.1** Define a B-Tree.

**Q 2.** What is a height balanced tree? Explain how the height is balanced after

addition/deletion of nodes in it?

**Q 3.** Write an algorithm to test whether a Binary Tree is a Binary Search Tree.

**Q.4** What are B-trees? Draw a B-tree of order 3 for the following sequence of

keys. 3,5,11,10,9,8,2,6,12 **(6)**

**Part B**

**Q.1** What is a Binary Search Tree (BST)? Make a BST for the following sequence

of numbers.

45, 36, 76, 23, 89, 116, 98, 39, 41, 56, 69, 48

Traverse the tree in Preorder, Inorder and postorder. **(8)**

**Q.2** Show the result of inserting the keys.

F, S, Q, K, C, L, H, T, V, W, M, R, N , P, A, B, X, Y, D, Z, E in the order to

an empty B-tree of degree-3. **(12)**

**Q.3** Make a BST for the following sequence of numbers.

45,32,90,34,68,72,16,24,30,66,11,50,10 Traverse the BST created in Preorder,

Inorder and Postorder. **(8)**

**Q4.** What are B-trees? Construct a B-Tree of order 3 for the following set of

Input data:

69, 19, 43, 16, 25, 40, 132, 100, 145, 7, 16, 18 **(7)**

**Q.33** Draw a B-tree of order 3 for the following sequence of keys:

2, 4, 9, 8, 7, 6, 3, 1, 5, 10 **(8)**

**Q.5** Explain insertion into a B-tree. **(8)**

**Q6.** Write an algorithm to delete a particular node from binary search tree. Trace

your algorithm to delete a node (10) from the given tree.

****

**Q7.** What is a Binary Search Tree (BST)? Make a BST for the following sequence

of numbers.

45, 32, 90, 21, 78, 65, 87, 132, 90, 96, 41, 74, 92 **(7)**

**Q8.** Traverse the Binary Search Tree created above in Preorder, Inorder and Postorder.**(7)**

**Miscellaneous Questions**

**Q.1** Write short notes on any **FOUR**:-

(i) B Tree.

(ii) Time Complexity, Big O notation.

(iii) Merge Sort.

(iv) Threaded Binary Tree.

(v) Depth First Traversal.

**Q.2** Write an algorithm INSERT that takes a pointer to a sorted list and a pointer to

a node and inserts the node into its correct position in the list. **(8)**

**Q. 3**Write short notes on the following:

(i) B-tree.

(ii) Abstract data type.

**Q.4** Define data type and abstract data type. Comment upon the significance of

both. **(8)**

**Q5** Enumerate various operations possible on ordered lists and arrays. Write

procedures to insert and delete an element in to array. **(8)**

**Q.6** By taking an example show how multidimensional array can be represented in

one dimensional array. **(8)**

**Q.7** Show the various passes of bubble sort on an unsorted list 11, 16, 2, 13, 6 **(8)**

**Q.8** Describe the concept of binary search technique? Is it efficient than sequential

search? **(8)**

**Q.9** Prove the hypothesis that “A tree having ‘m’ nodes has exactly (m–1) edges or

branches”. **(8)**

****

**Q.10** Write a procedure to insert a node into a linked list at a specific position and

draw the same by taking any example? **(8)**

**Q.11** List various problem solving techniques. **(5)**

**Q12.** Explain the concept of primitive data structures. **(4)**

**Q13.** The system allocates memory for any multidimensional array from a large

single dimensional array. Describe *two* mapping schemes that helps us to store

a *two* dimensional metrics in a *one-dimensional* array. **(8)**

**Q14.** Write an algorithm for binary search. What are the conditions under which

sequential search of a list is preferred over binary search? **(7)**

**Q16.** Define the following terms:

(i) Abstract data type.

(ii) Column major ordering for arrays.

(iii) Adjacency multilist.

(iv) Game trees. **(14)**

**Q16.** Describe various memory allocation strategies. **(8)**

**Q17.** How memory is freed using Boundary tag method in the context of Dynamic

memory management? **(6)**

**Q 18.** Define a method for keeping two stacks within a single linear array S in such a

way that neither stack overflows until entire array is used and an entire stack is

never shifted to a different location within the array. Write routines for pushing

and poping elements in two stacks. **(8)**

**Q19.** Suppose a queue is housed in an array in circular fashion. It is desired to add

new items to the queue. Write down a procedure ENQ to achieve this also

checking whether the queue is full. Write another procedure DQ to delete an

element after checking queue empty status. **(6)**

**Q20.** Write short notes on the following:

(i) Threaded binary trees.

(ii) Graph traversal.

(iii) Conversion of forest into tree.

(iv) Doubly linked list. **( 3.5****4** **14 )**

**Q21.** Differentiate between system defined data types and Abstract data types with

suitable examples. **(8)**

**Q22.** Explain the following:

(i) Complexity of an Algorithm.

**(ii)** The space-time trade off algorithm. **(6)**

**Q23.** Let a binary tree ‘T’ be in memory. Write a procedure to delete all terminal

nodes of the tree. **(8)**

**Q24.** Consider the following eight numbers 50, 33, 44, 22, 77, 35, 60 and 40. Display

the construction of the binary by inserting the above numbers in the given order.**(6)**

**Q25.** Establish the usage of linked lists for polynomial manipulation. **(5)**

**Q26.** Define a linked-list? How are these stored in the memory? Suppose the linked

list in the memory consisting of numerical values. Write a procedure for each of

the following:

(i) To find the maximum MAX of the values in the list.

(ii) To find the average MEAN of the values in the list.

(iii) To find the product PROD of the values in the list. **(14)**

**Q27.** Give the binary search algorithm. **(7)**

**Q28.** What do you understand by structured programming? Explain. **(5)**

**Q29.** Consider the algebraic expression

E = (5x+z) (3a-b)2

(i) Draw the expression tree corresponding to E

(ii) Find the scope of exponential operator i.e. the subtree rooted at the

exponential operator. (7)

**Q30.** Define an array. How does an array differ from an ordinary variable? How are

arrays represented in the memory? **(5)**

**Q31.** Consider an array A[20, 10]. Assume 4 words per memory cell and the base

address of array A is 100. Find the address of A[11, 5] assuming row major

storage. **(5)**

**Q32.** Write a recursive function to count the number of nodes in a binary tree. **(7)**

**Q33.** Define the following :

(i) AVL tree.

(ii) Thread.

(iii) Heap.

(iv) Binary Search Tree. **(8)**

**Q34.** Write an algorithm for searching a key from a sorted list using binary search

technique. **(6)**

**Q35.** Define graph, adjacency matrix, adjacency list, hash function, sparse matrix,

reachability matrix. **(6)**

**Q36.** Explain various graph traversal schemes and write their merits and demerits.**(8)**

**Q37.** Write short notes on the following:

(i) Decision and game trees.

(ii) Polynomial representation and manipulation using linked lists.

(iii) Analysis of algorithm.

(iv) Circular queues. **(3 ½ x 4 = 14)**

**Q38.** What is the smallest value of n such that an algorithm whose running time is

100n2 runs faster than an algorithm whose running time is 2n on the same

machine. **(4)**

**Q39.** Let X = (X1, X2, X3,….Xn) and Y= (Y1, Y2, Y3,….Xm) be two linked lists.

Write an algorithm to merge the lists together to obtain the linked list Z such that

Z = (X1, Y1, X2, Y2,….Xm, Ym,Xm+1….Xn) if m<=n or

Z = (X1, Y1,X2,Y2….Xn,Yn,Yn+1….Ym) if m>n. **(7)**

**Q40.** Devise a representation for a list where insertions and deletions can be made at

either end. Such a structure is called a Deque (Double ended queue). Write

functions for inserting and deleting at either end. **(7)**

**Q41.** Write binary search algorithm and trace to search element 91 in following list:

13 30 62 73 81 88 91

What are the limitations of Binary Search? **(7)**

**Q42.** Show the result of running BFS on a complete Binary Tree of depth 3. Show the

status of the data-structure used at each stage. **(6)**

**Q43.** Define a linked list with a loop as a linked list in which the tail element points

to one of the list’s elements and not to NULL. Assume that you are given a

linked list *L*, and two pointers *P1*, *P2* to the head. Write an algorithm that

decides whether the list has a loop without modifying the original list. The

algorithm should run in time *O(n)* and additional memory *O(1)*, where *n* is the

number of elements in the list. **(10)**

Q44 Write an algorithm for checking validity of the input, i.e., the program must

know if the input is disjoint, duplicated and has a loop. **(10)**

**Q.45** Write an algorithm for finding solution to the Tower’s of Hanoi problem.

Explain the working of your algorithm (with 4 disks) with diagrams.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

**Assignment Questions**

**Unit 1**

1. a) Explain the representation of doubly linked list.

b) Write about the operations performed on doubly linked list.

1. Define algorithm. What are the desirable properties of the best algorithm.
2. Add the following operation to the Natural Number ADT: Predecessor, IsGreater, Multiply, Divide.
3. An algorithm takes 0.5ms for input size 100.How long will it take for input size 500 if the running time is the following:
4. Linear b)O(Nlog N) c)Quadratic d) Cubic
5. Show that the following statements are correct
   * 1. 5n2-6n=ϴ(n2) b) 2n2+nlogn= ϴ(n2)
6. How do you find the complexity of an algorithm? What is the time and space complexity of an algorithm? Justify your answer with an example.
7. Suppose , an array A[-16,…64] is stored in a memory whose starting address is 459. Assume that word size for each element is 2. Then obtain the following

i). How many no. of elements are there in the array A.

ii). If one word of the memory is equal to 2 bytes, then how much memory is required to store the entire array.

iii). What is the location for A[50].

iv). What is the location for 10th element?

v). which element is located at 589.

8. Given a singly linked list ,write a function to swap every two nodes eg:1->2->3->4->5->6 should become 2->1->4->3->6->5.

**Unit -2**

**1.** Convert the following infix expression into a postfix expression (Show steps)

(i)ABD/ E −FG H/ k

(ii) A B D/E −FG

(iii) a bc d/e f g .

2 After obtaining the postfix expression on the above expressions reverse them into infix expression using corresponding algorithm

**Unit 3**

1. A Binary tree has 9 nodes. The inorder and postorder traversals of the tree

yields the following sequence of nodes:

Inorder : 1 2 3 4 5 6 7 8 9

Postorder: 1 3 5 4 2 8 7 9 6

Draw the tree. Explain your algorithm.

1. How will you represent a max-heap sequentially? Explain with an example.
2. Construct the binary tree for the following sequence of nodes in preorder and

inorder respectively.

Preorder : G, B, Q, A, C, K, F, P, D, E, R, H

Inorder: Q, B, K, C, F, A, G, P, E, D, H, R

1. Give the algorithm to construct a binary tree where the yields of preorder and

post order traversal are given.

1. Draw a picture of the directed graph specified below:

G = ( V, E)

V(G) = {1, 2, 3, 4, 5, 6}

E(G) = {(1,2), (2, 3), (3, 4), (5,1), (5, 6), (2, 6), (1, 6), (4, 6), (2, 4)}

Obtain the following for the above graph:

* 1. Adjacency matrix.
  2. React ability matrix.
  3. **A**djacency List.

**Unit 4**

1. What is quick sort? Sort the following array using quick sort method.

24 56 47 35 10 90 82 31 **(7)**

1. Sort the following sequence of keys using merge sort.

66, 77, 11, 88, 99, 22, 33, 44, 55 **(8)**

1. Apply radix sort ,insertion sort, selection sort techniques on the above data
2. The following values are to be stored in a hash table

25, 42, 96, 101, 102, 162, 197

Describe how the values are hashed by using division method of hashing with

a table size of 7. Use chaining as the method of collision resolution. **(8)**

**Unit 5**

1. What are B-trees? Draw a B-tree of order 3 for the following sequence of

keys. 3,5,11,10,9,8,2,6,12 **(6)**

1. What is a Binary Search Tree (BST)? Make a BST for the following sequence

of numbers.45, 32, 90, 21, 78, 65, 87, 132, 90, 96, 41, 74, 92 **(7)**

1. construct an AVL tree for the following data
   * 1. 30,31,32,23,22,28,24,29,26,27,34,36
     2. 50,55,60,16,20,40,20,45,30,70,80
2. Write short notes Pattern Matching Algorithms
3. Write short notes Tries
4. Explain Red –black trees
5. Explain Splay trees.

**Objective Questions**

**Unit 1**

1) The asymptotic analysis focuses on determining \_\_\_\_\_\_\_terms in the complexity function

2) The data space is needed store\_\_\_\_\_\_\_\_\_

3) Consider a linked list of n elements. What is the time taken to insert an element pointer ? [ ]

1. O(log2n) B. O(n) C.O(1) D.O(n log2n)

4) Data that consists of a single, non decomposable entity are known [ ]

(A) atomic data (B) array new (C) data structure delete (D) standard type

**Unit 2**

1) Which of the following operation is used to add an item in a queue [ ]

(A) write() (B) read() (C) pop() (D) push()

2) Queue can be used to implement [ ]

(A) recursion (B) quick sort (C) radix sort (D) depth first search

**Unit 3**

1. A priority queue can be implemented by [ ]

a) Heap b) BST c) DFS method d) AVL Tree

2. The difference between tree and graph will be [ ]

a) Tree has no cycles, graph can have cycle b) Tree has no parent, graph can have parent

c) Tree has root node, graph has no root node d) Both A and C

3. Which of the following is useful in traversing a given graph by breadth first search [ ]

a) Stack s b) Set c) list d) Queue

4. In a heap\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ element will resides at top position.

5. In a max heap the child element should be \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ than parent element

6. if a heap represented in the form of list, when a parent element available at “ ith “ element in the list then left, right Childs will be available at \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

7. The post order traversal of a binary tree is DEBFCA. find out the pre order traversal [ ]

A)ABFCDE B)ADBFEC C)ABDECF D)ABDCEF

8. The time to initialize the max heap is \_\_\_\_\_\_\_

9) The data structure that is used to keep the vertices whose adjacent vertices are to be visited in the Depth first traversal\_\_\_\_\_\_\_\_\_\_\_ [ ]

a) Queue b) stack c) heap d) dictionary.

10) The number of edges incident from a vertex vi called\_\_\_\_\_\_\_ [ ]

a) In degree b) out degree c) pendent d) degree

11) Graph can be represented by [ ]

a) Adjacency matrix b) adjacency list c) queue d) both a & b

12) \_\_\_\_\_\_\_\_\_ is the application of Priority queue [ ]

a) Scheduling of jobs in operating system b)text editors

c)spell checking programs d)heap

13) Graph is a collection of \_\_\_\_\_\_\_\_\_\_ and \_\_\_\_\_\_\_\_\_\_\_

14) \_\_\_\_\_\_\_\_\_\_\_ is required when data being sorted do not fit in to main memory.

16) \_\_\_\_\_\_ consists of a set of vertices V and a set of edges E.

16) \_\_\_\_\_\_\_is a complete binary tree in which the value in each node is lesser than those in its children.

17) In an undirected graph, the sum of degrees of all the nodes [ ]

(A) must be even (B) is thrice the number of edges

(C) must be odd (D) need not be even

18) The minimum number of edges in a connected cyclic on n vertices is \_\_\_\_\_\_\_\_\_\_\_\_\_

19) An n vertex undirected graph with exactly n\*(n-1)/2 edges is said to be \_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Unit 4**

1) The order of the binary search algorithm is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2) In hashing by division the hash function has the form\_\_\_\_\_ [ ]

a)f(k)=K%(d-1) b) f(k)=(K+1)%(d+1) c)f(k)=(k-1)%d d)f(k)=k%d

3) In division hash function , in the hash table of length 11 we can place the value 80 at \_\_\_\_\_position. [ ]

a)5 b)8 c)3 d)10

4)\_\_\_\_\_\_\_\_\_\_ occurs when there isn't room i n the home bucket for the new pair

5) One of the collision handling method \_\_\_\_\_\_\_\_

6) A -------------- sort uses the binary tree concept such that any number is larger than all the numbers in the subtree below it is called [ ]

(A) Quick (B) Bubble (C) Heap t (D) All

7) The number of passes required for sorting M records of length N using simple external sorting algorithm is [ ]

(A) [log(N/M)] (B) [log(M/N)] (C) [log(N\*M)] (D) [log(N+M)]

8) For merging two sorted lists of sizes m and n into a sorted list of size m+n, requires \_ \_\_ \_ \_ \_ \_ \_ no.of comparisons. [ ]

a) O(m) b) O(n) c) O(m+n) d) O(log(m)+log(n))

9) Sorting is not useful for [ ]

a) report generation b) minimizing the storage needed

c) making searching easier and efficient d) responding to queries easily

10) Merge sort uses\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ [ ]

a)divide and conquer b)backtracking c)greedy approach d)heuristic approach

11) The average number of comparisons performed by the merge sort algorithm , in merging two sorted listsof length 2 is [ ]

(A) 8/3 (B) 8/5 (C) 11/7 (D) 1/16

**Unit 5**

1) A binary search tree contains the values - 1,2,3,4,5,6,7,8. The tree is traversed in preorder and the values are printed out. Which of the following sequences is a valid output? [ ]

(A) 5 1 2 3 (B) 1 4 2 6 (C) 1 2 3 4 (D) 5 3 1 2

2) In a binary search tree if the key element is less than the root element then sub tree must be searched in \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ traversal of a binary search tree traverses visits to the nodes in ascending order of key values.

4). In a BST, parent element should be [ ]

a) <left,>right b) >left, <right c) <left, <right d) >left, >right

5) In a red black tree, a root node is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ leaf node is \_\_\_\_\_\_\_\_\_ [ ]

a) Black, Red b) Red, Black c) Black, Black d) Red, Red

6) Which of the following is search engine [ ]

a) BST b) AVL tree c) Brute Force Alg d) Splay tree

7) A node in a B-tree consists of set of elements, those should be arranged in [ ]

a) Non-increasing order b) Non-decreasing order c) Both, depends on data d) None

8) In an AVL tree, the heights of left, right sub child are differed by [ ]

a) At most one b) At least one c) One d) Depends on data

9) Suffix Trie search time [ ]

a) O (n+1) b) O (n-1) c) O (m) d) None

10) The order of B-tree indicates [ ]

a) Number of element present in the tree b) Number of element present at certain node

c) Total number of leaf nodes d) None

11) In a BST the left child should be \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ than parent element

12) AVL tree is also known as \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

13) A height of BST will be performed by \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ traversal

14) When an element inserted at left sub tree of left child then \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ rotation will be performed.

16) The Knuth-Morris-Pratt (KMP) algorithm looks for the pattern in the text in a \_\_\_\_\_\_\_\_\_\_\_\_order

16) A compressed trie as internal nodes of degree al least \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

17) In a binary tree, certain null entries are replaced by special pointers which point to nodes higher in the tree for efficiency. These special pointers are called \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

18) A binary search tree is constructed with the following keys 20,22,26,21,13,19,18,16,26,28

The above keys are inserted in that order. Then the total keys in the left sub tree and the right sub tree of the tree or respectably. [ ]

a) 5,5 b) 6,4 c) 7,3 d) 4,6

19) The balance factor of a node x in a binary tree is 3. There are 2 nodes in the right sub tree of x. There must be \_ \_ \_ \_ \_ \_ \_ \_ nodes in the left sub tree of x [ ]

a) 2 b) 0 c) 5 d) 3

20) AVL tree is a \_ \_ \_ \_ \_ \_ \_ \_ \_ binary tree [ ]

a) Complete b) Full c) Height balanced d) Skewed

21) In KMP pattern matching algorithm pre processing is done by an auxillary function known as

a) failure function b) prefix function c) postfix function d) insert function

22) In R0 Rotation the node which is imbalanced will be moving towards [ ]

a) Left sub tree b) Right Subtree c) root d) not moving

23) Which traversal of a binary search tree traverses visits to the nodes in ascending order of key values?

a) In Order b) Pre Order c) Post Order d) Past Order

24) AVL tree was not developed by \_ \_ \_ \_ \_ \_ \_ \_ \_ [ ]

a) Velskii b) Anderson c) Landis d) Adelson

25) B-tree of order 3 is also known as \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ [ ]

a) 2-3 tree b) 3-4 tree c) binary tree d) Splay tree

26) In binary search tree, if the element to be inserted is greater than the root node, the element is inserted in \_\_\_\_\_\_\_\_\_\_\_\_

27. The difference between the height of left sub tree & right sub tree is called \_\_\_\_\_\_\_\_

28. All AVL Trees are basically \_\_\_\_\_\_\_\_\_\_

29. \_\_\_\_\_\_\_\_\_ algorithm is recommended for binary strings pattern matching.

30. The permissible balance factors of an AVL trees are \_\_\_\_\_\_\_\_\_

31. In B-tree of order m, the root has child nodes\_\_\_\_\_\_\_\_\_\_\_

32. \_\_\_\_\_\_\_\_\_\_\_ traversal technique the node is processed before the children.

33. \_\_\_\_\_\_\_\_ Tress is designed especially for use on disk.

34. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ algorithm is preferred for pattern matching when the length is of short duration.

35 In a B-tree of order ‘m’ all the leaf nodes except the root node should have minimum of \_\_\_ non empty children. [ ]

a) [m/2] b) [m] c) [m-1] d) [(m/2)-1]

36 \_\_\_\_\_\_\_ algorithm is recommended for the binary strings pattern matching [ ]

a)Brute force b) Boyer Moore c)KMP d)Morris

37 In LR Rotation the node which is imbalanced is replaced by\_\_\_\_\_\_\_\_\_\_\_\_ [ ]

a) root of the left subtree b)root of right subtree

c) left child of right subtree d) right child of left subtree

38 A compressed trie is a kind of standard trie in which internal node has atleast degree of [ ]

a) 3 b) 1 c) 2 d)-1

39) The difference between heights of left subtree and Right subtree is called [ ]

a) Balanced factor b) height difference c) Rank d) Load balance

40) In a AVL Tree LL rotation the node which is imbalanced will move towards\_\_\_\_\_\_\_\_\_\_

41) The search, insert and delete operations on a m-way search tree of height have the complexity as \_\_\_\_\_\_

42) \_\_\_\_\_\_\_\_\_\_algorithm is preferred for pattern matching if the size of string is large compared to the length of the pattern.

43) A node with ‘k’ subtrees will have\_\_\_\_\_\_ elements in B-Tree.

44) \_\_\_\_\_\_\_\_\_ is a technique of finding the substring in text which is equal to pattern.

45) \_\_\_\_\_\_\_\_\_\_ is collection of elements that each element has been added a priority.

46) Find the odd one out [ ]

(A)binary tree (B) AVL tree (C) graph (D) queue

47) Which of the following need not be a binary tree [ ]

(A)Search tree (B) Heap (C) AVL-tree (D) B-tree

48) Which of the following traversal techniques lists the nodes of a binary search tree in ascending order

(A) post-order (B) In-order (C) Pre-order (D) No-order

49) In which trie node allows only one character [ ]

(A) standard (B) compressed (C) suffix (D) none

50) A priority queue is an abstract concept like a \_\_\_\_\_\_\_\_\_\_\_\_

51) Merge sort uses\_\_\_\_\_\_\_\_\_\_\_\_\_ strategy

52**)** The depth of a complete binary tree with n nodes **\_\_\_\_\_\_\_\_\_\_**

53) \_\_\_\_\_\_\_\_\_\_ tree is a self-balancing binary search tree

54) A standard trie uses\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ space

55) A \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ is a tree-based data structure, stores the large text string as a tree for fast pattern matching.

**Tutorial Problems**

**UNIT-I**

* + 1. Add the following operation to the Natural Number ADT: Predecessor, IsGreater, Multiply, Divide.
    2. Create an ADT set: use the standard mathematical definition and include the following operations: Crate, Insert, Remove, IsIn, Union, Intersection and Difference.
    3. Show that the following statements are correct
    4. 5n2-6n=ϴ(n2) b) 2n2+nlogn= ϴ(n2)

1. Show that the following statements are incorrect.
   * 1. 10n2+9=O(n) b) n2logn= ϴ(n2)
2. Suppose , an arrayA[-16,…64] is stored in a memory whose starting address is 459. Assume that word size for each element is 2. Then obtain the following

i). How many no. of elements are there in the array A.

ii). If one word of the memory is equal to 2 bytes, then how much memory is required to store the entire array.

iii). What is the location for A[50].

iv). What is the location for 10th element?

v). which element is located at 589.

**UNIT-II**

* 1. Transform the following infix expressions into their equivalent postfix expressions

i). A\*(B+D)/F-F\*(G+H/K)

ii). A^B\*C+D/A/(E+F)

**UNIT-III**

1. Draw the internal memory representations of the binary tree using

a) Sequential and b) Linked Representation.

****

2. Write the preorder, inorder and post order traversals of the binary tree given in Q.1

3. Draw the binary tree given in Q.1, showing its threaded representation.

4. Suppose that we have the following key values7,16,49,82,5,31,6,2,44

a). Write out the max heap after each value is inserted into the heap.

b). Write the min heap after each value is inserted into the heap.

5. Consider the following specification of a graph G, V(G)={1,2,3,4}

E(G)={(1,2),(1,3),(3,3),(3,4),(4,1)}

a). Draw a picture of the undirected graph.

b). Give its Adjacency Matrix.

c). Give its Adjacency List.

d). Write BFS and DFS Traversals.

**UNIT-IV**

* 1. Consider the given unsorted array. Sort this array in ascending order using insertion sort.76, 67, 36, 55, 23, 14, 6
  2. Sort this array using selection sort and show your work in each pass.

7, 23,31,40,56, 78,92

* 1. Sort 07, 10, 99, 02, 80, 14, 25, 63, 88, 33, 11, 72, 68, 39,21, 50 using Radix Sort.
  2. Sort the following numbers using Quick Sort.

3, 1,4,5,9,2,6,10,7,8

1. Given the input{4371, 1323,6173,4199,4344,9699,1889} and hash function as

Key%10. Show the results for the following.

* + - 1. Open addressing using linear probing.
      2. Open addressing using quadratic probing.
      3. Open addressing using double hashing.

**UNIT-V**

* 1. Construct the AVL Tree for the following data.
  2. Explain the steps to build a B-Tree of order 3 for the following data.
  3. Explain how Brute- Force algorithm searches for abdf in pattern abdadefg.
  4. Apply Knuth-Morris –Pratt algorithm to P=bacaaa and

T=bacbacabcbacbbbacabacbabcbbba

5 . Construct a trie for the set of keywords={inner,input,in,outer,output,put,outing,tint}

**Known Gaps**

Fortunately, no known gaps as it is extension of C Programming in their I Year.

**Discussion Topics**

* Types of Linked Lists
* Sparse matrices representation and manipulation.
* Queues- Circular Queues, Double Ended Queues.
* Non Recursive Binary Tree Traversals.
* Searching Strategies- Linear and Binary Search, Hashing.
* Comparison of Sorting techniques.
* Search Trees.
* Linked list implementation of various Data Structures.

Pattern Matching Algorithms.