DMI COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERI NG

CS8391-DATA STRUCTURES QUESTION BANK

UNIT I

2MARKS

1.Explain the term data structure.

The data structure can be defined as the collection of elements and all the possible operations which are required for those set of elements. Formally data structure can be defined as a data structure is a set of domains D, a set of domains F and a set of axioms A. this triple (D,F,A) denotes the data structure d.

2. What do you mean by non-linear data structure? Give example.

The non-linear data structure is the kind of data structure in which the data may be arranged in hierarchical fashion. For example- Trees and graphs.

3. What do you linear data structure? Give example.

The linear data structure is the kind of data structure in which the data is linearly arranged. For example- stacks, queues, linked list.

4. Enlist the various operations that can be performed on data structure.

Various operations that can be performed on the data structure are

- Create
- Insertion of element
- Deletion of element
- Searching for the desired element
- Sorting the elements in the data structure
- Reversing the list of elements.

5. What is abstract data type? What are all not concerned in an ADT?
6.

The abstract data type is a triple of D i.e. set of axioms, F-set of functions and A-Axioms in which only what is to be done is mentioned but how is to be done is not mentioned. Thus ADT is not concerned with implementation details.

6. List out the areas in which data structures are applied extensively.

Following are the areas in which data structures are applied extensively.

- Operating system- the data structures like priority queues are used for scheduling the jobs in the operating system.
- Compiler design- the tree data structure is used in parsing the source program. Stack data structure is used in handling recursive calls.
- Database management system- The file data structure is used in database management systems. Sorting and searching techniques can be applied on these data in the file.
- Numerical analysis package- the array is used to perform the numerical analysis on the given set of data.
- Graphics- the array and the linked list are useful in graphics applications.
- Artificial intelligence- the graph and trees are used for the applications like building expression trees, game playing.

7. What is a linked list?

A linked list is a set of nodes where each node has two fields 'data' and 'link'. The data field is used to store actual piece of information and link field is used to store address of next node.

8. What are the pitfall encountered in singly linked list?

Following are the pitfall encountered in singly linked list

- The singly linked list has only forward pointer and no backward link is provided. Hence the traversing of the list is possible only in one direction. Backward traversing is not possible.
- Insertion and deletion operations are less efficient because for inserting the element at desired position the list needs to be traversed. Similarly, traversing of the list is required for locating the element which needs to be deleted.

9. Define doubly linked list.

Doubly linked list is a kind of linked list in which each node has two link fields. One link field stores the address of previous node and the other link field stores the address of the next node.

10. Write down the steps to modify a node in linked lists.

- ➢ Enter the position of the node which is to be modified.
- ➢ Enter the new value for the node to be modified.
- ➢ Search the corresponding node in the linked list.
- ➢ Replace the original value of that node by a new value.
- ➢ **Display the messages as " the node is modified".**

11. Difference between arrays and lists.

In arrays any element can be accessed randomly with the help of index of array, whereas in lists any element can be accessed by sequential access only.

Insertion and deletion of data is difficult in arrays on the other hand insertion and deletion of data is easy in lists.

12. State the properties of LIST abstract data type with suitable example.

Various properties of LIST abstract data type are

(i) It is linear data structure in which the elements are arranged adjacent to each other.

(ii) It allows to store single variable polynomial.

(iii)If the LIST is implemented using dynamic memory then it is called linked list. Example of LIST are- stacks, queues, linked list.

13. State the advantages of circular lists over doubly linked list.

In circular list the next pointer of last node points to head node, whereas in doubly linked list each node has two pointers: one previous pointer and another is next pointer. The main advantage of circular list over doubly linked list is that with the help of single pointer field we can access head node quickly. Hence some amount of memory get saved because in circular list only one pointer is reserved.

14. What are the advantages of doubly linked list over singly linked list?

The doubly linked list has two pointer fields. One field is previous link field and another is next link field. Because of these two pointer fields we can access any node efficiently whereas in singly linked list only one pointer field is there which stores forward pointer.

15. Why is the linked list used for polynomial arithmetic?

We can have separate coefficient and exponent fields for representing each term of polynomial. Hence there is no limit for exponent. We can have any number as an exponent.

16. What is the advantage of linked list over arrays?

The linked list makes use of the dynamic memory allocation. Hence the user can allocate or de allocate the memory as per his requirements. On the other hand, the array makes use of the static memory location. Hence there are chances of wastage of the memory or shortage of memory for allocation.

17. What is the circular linked list?

The circular linked list is a kind of linked list in which the last node is connected to the first node or head node of the linked list.

18. What is the basic purpose of header of the linked list?

The header node is the very first node of the linked list. Sometimes a dummy value such -999 is stored in the data field of header node.

This node is useful for getting the starting address of the linked list.

19. What is the advantage of an ADT?

- **Change:** the implementation of the ADT can be changed without making changes in the client program that uses the ADT.
- **Understandability:** ADT specifies what is to be done and does not specify the implementation details. Hence code becomes easy to understand due to ADT.
- **Reusability:** the ADT can be reused by some program in future.

20. What is static linked list? State any two applications of it.

- The linked list structure which can be represented using arrays is called static linked list.
- It is easy to implement, hence for creation of small databases, it is useful.
- The searching of any record is efficient, hence the applications in which the record need to be searched quickly, the static linked list are used.

16 MARKS

1. Explain the insertion operation in linked list. How nodes are inserted after a specified node.

2. Write an algorithm to insert a node at the beginning of list?

3. Discuss the merge operation in circular linked lists.

4. What are the applications of linked list in dynamic storage management?

5. How polynomial expression can be represented using linked list?

6. What are the benefit and limitations of linked list?

7. Define the deletion operation from a linked list.

8. What are the different types of data structure?

9. Explain the operation of traversing linked list. Write the algorithm and give an

example.

UNIT II

2MARKS

1. Define Stack

A Stack is an ordered list in which all insertions (Push operation) and deletion (Pop operation) are made at one end, called the top. The topmost element is pointed by top. The top is initialized to -1 when the stack is created that is when the stack is empty. In a stack S = (a1,an), a1 is the bottom most element and element a is on top of element ai-1. Stack is also referred as Last In First Out (LIFO) list.

2. What are the various Operations performed on the Stack?

The various operations that are performed on the stack are

CREATE(S) – Creates S as an empty stack.

PUSH(S,X) – Adds the element X to the top of the stack.

POP(S) – Deletes the top most elements from the stack.

TOP(S) – returns the value of top element from the stack.

ISEMTPTY(S) – returns true if Stack is empty else false.

ISFULL(S) - returns true if Stack is full else false.

3.How do you test for an empty stack?

The condition for testing an empty stack is top =-1, where top is the pointer pointing to the topmost element of the stack, in the array implementation of stack. In linked list implementation of stack the condition for an empty stack is the header node link field is NULL.

4.Name two applications of stack?

Nested and Recursive functions can be implemented using stack. Conversion of Infix to Postfix expression can be implemented using stack. Evaluation of Postfix expression can be implemented using stack.

5.Define a suffix expression.

The notation used to write the operator at the end of the operands is called suffix notation.

Suffix notation format : operand operand operator

Example: ab+, where a & b are operands and '+' is addition operator.

6.What do you meant by fully parenthesized expression? Give example.

A pair of parentheses has the same parenthetical level as that of the operator to which it corresponds. Such an expression is called fully parenthesized expression.

Ex: (a+((b*c) + (d * e))

7.Write the postfix form for the expression -A+B-C+D?

A-B+C-D+

8.What are the postfix and prefix forms of the expression?

A+B*(C-D)/(P-R)

Postfix form: ABCD-*PR-/+

Prefix form: +A/*B-CD-PR

9.**Explain the usage of stack in recursive algorithm implementation?**

In recursive algorithms, stack data structures is used to store the return address when a recursive call is encountered and also to store the values of all the parameters essential to the current state of the function.

10.**Define Queues.**

A Queue is an ordered list in which all insertions take place at one end called the rear, while all deletions take place at the other end called the front. Rear is initialized to -1 and front is initialized to 0. Queue is also referred as First In First Out (FIFO) list.

11.**What are the various operations performed on the Queue?**

The various operations performed on the queue are

CREATE(Q) – Creates Q as an empty Queue.

Enqueue(Q,X) – Adds the element X to the Queue.

Dequeue(Q) – Deletes a element from the Queue.

ISEMTPTY(Q) – returns true if Queue is empty else false.

ISFULL(Q) - returns true if Queue is full else false.

12. **How do you test for an empty Queue?**

The condition for testing an empty queue is rear=front-1. In linked list implementation of queue the condition for an empty queue is the header node link field is NULL.

13.**Write down the function to insert an element into a queue, in which the queue is implemented as an array. (May 10)**

Q – Queue

X – element to added to the queue Q IsFull(Q) –

Checks and true if Queue Q is full Q->Size -

Number of elements in the queue Q Q->Rear –

Points to last element of the queue Q Q->Array

– array used to store queue elements void

```
enqueue (int X, Queue Q) {
        if(IsFull(Q))
          Error ("Full queue");
        else      {
                Q->Size++;
                    Q->Rear = Q->Rear+1;
                    Q->Array[ Q->Rear ]=X;
                }
        }
```

14.Define Dequeue.

Deque stands for Double ended queue. It is a linear list in which insertions and deletion are made from either end of the queue structure.

15.**Define Circular Queue.**

Another representation of a queue, which prevents an excessive use of memory by arranging elements/ nodes Q_1,Q_2,**…**Q_n in a circular fashion. That is, it is the queue, which wraps around upon reaching the end of the queue

16 MARKS

1. Write an algorithm for Push and Pop operations on Stack using Linked list. (8)
2. Explain the linked list implementation of stack ADT in detail?
3. Define an efficient representation of two stacks in a given area of memory with n words and explain.

4. Explain linear linked implementation of Stack and Queue?
    a. Write an ADT to implement stack of size N using an array. The elements in the stack are to be integers. The operations to be supported are PUSH, POP and DISPLAY. Take into account the exceptions of stack overflow and stack underflow. (8)
    b. A circular queue has a size of 5 and has 3 elements 10,20 and 40 where F=2 and R=4. After inserting 50 and 60, what is the value of F and R. Trying to insert 30 at this stage what happens? Delete 2 elements from the queue and insert 70, 80 & 90. Show the sequence of steps with necessary diagrams with the value of F & R. (8 Marks)
5. Write the algorithm for converting infix expression to postfix (polish) expression?
6. Explain in detail about priority queue ADT in detail?
7. **Write a function called 'push' that takes two parameters: an integer variable and a stack into** which it would push this element and returns a 1 or a 0 to show success of addition or failure.
8. What is a DeQueue? Explain its operation with example?
9. Explain the array implementation of queue ADT in detail?
10. Explain the addition and deletion operations performed on a circular queue with necessary algorithms.(8) (Nov 09)

UNIT III

2MARKS

1. **Define tree?**

   Trees are non-liner data structure, which is used to store data items in a shorted sequence.

   It represents any hierarchical relationship between any data Item. It is a collection of nodes, which has a distinguish node called the root and zero or more non-empty sub trees T1, **T2,….Tk. each of which are connected by a directed edge from the root.**

2. Define Height of tree?

The height of n is the length of the longest path from root to a leaf. Thus all leaves have height zero. The height of a tree is equal to a height of a root.

3. Define Depth of tree?

For any node n, the depth of n is the length of the unique path from the root to node n. Thus for a root the depth is always zero.

4. What is the length of the path in a tree?

The length of the path is the number of edges on the path. In a tree there is exactly one path form the root to each node

5. Define sibling?

Nodes with the same parent are called siblings. The nodes with common parents are called siblings.

6. Define binary tree?

A Binary tree is a finite set of data items which is either empty or consists of a single item called root and two disjoin binary trees called left sub tree max degree of any node is two.

7. What are the two methods of binary tree implementation?

Two methods to implement a binary tree are,

a. Linear representation.

b. Linked representation

8. What are the applications of binary tree?

Binary tree is used in data processing.

a. File index schemes

b. Hierarchical database management system

9. List out few of the Application of tree data-structure?

Ø The manipulation of Arithmetic expression

Ø Used for Searching Operation

Ø Used to implement the file system of several popular operating systems

Ø Symbol Table construction

Ø Syntax analysis

10. Define expression tree?

Expression tree is also a binary tree in which the leafs terminal nodes or operands and non-terminal intermediate nodes are operators used for traversal.

11. Define in -order traversal?

In-order traversal entails the following steps;

a. Traverse the left subtree
b. Visit the root node
c. Traverse the right subtree

13. Define threaded binary tree.

A binary tree is threaded by making all right child pointers that would normally be null point to the in order successor of the node, and all left child pointers that would normally be null point to the in order predecessor of the node.

14. What are the types of threaded binary tree?

i. Right-in threaded binary tree
ii. Left-in threaded binary tree
iii. Fully-in threaded binary tree

15. Define Binary Search Tree.

Binary search tree is a binary tree in which for every node X in the tree, the values of all the keys in its left subtree are smaller than the key value in X and the values of all the keys in its right subtree are larger than the key value in X.

16.What is AVL Tree?

AVL stands for Adelson-Velskii and Landis. An AVL tree is a binary search tree which has the following properties:

1.The sub-trees of every node differ in height by at most one.
2.Every sub-tree is an AVL tree.
Search time is O(logn). Addition and deletion operations also take O(logn) time.

17. List out the steps involved in deleting a node from a binary search tree.

- Deleting a node is a leaf node (ie) No children
- Deleting a node with one child.
- Deleting a node with two Childs.

18. What is binomial heaps?

A binomial heap is a collection of binomial trees that satisfies the following binomial-heap properties:

1. No two binomial trees in the collection have the same size.

2. Each node in each tree has a key.

3. Each binomial tree in the collection is heap-ordered in the sense that each

   non-root has a key strictly less than the key of its parent

.The number of trees in a binomial heap is O(log n).

20. Define complete binary tree.

If all its levels, possible except the last, have maximum number of nodes and if all the nodes in the last level appear as far left as possible.

16 ARKS

1.Explain the AVL tree insertion and deletion with suitable example.
2.Describe the algorithms used to perform single and double rotation on AVL tree.
3.Explain about B-Tree with suitable example.
4.Explain about B+ trees with suitable algorithm.
5.Write short notes on
   i. Binomial heaps
   ii. Fibonacci heaps
6.Explain the tree traversal techniques with an example.
7.Construct an expression tree for the expression (a+b*c) + ((d*e+f)*g). Give the outputs when you apply inorder, preorder and postorder traversals.
8.How to insert and delete an element into a binary search tree and write down the code for the insertion routine with an example.
9.What are threaded binary tree? Write an algorithm for inserting a node in a threaded binary tree.
10.Create a binary search tree for the following numbers start from an empty binary search tree. 45,26,10,60,70,30,40 Delete keys 10,60 and 45 one after the other and show the trees at each stage.

UNIT- IV

PART A

1. **Write the definition of weighted graph?**
   A graph in which weights are assigned to every edge is called a weighted graph.

2. **Define Graph?**
   A graph G consist of a nonempty set V which is a set of nodes of the graph, a set E which is the set of edges of the graph, and a mapping from the set of edges E to set of pairs of elements of V. It can also be represented as G=(V, E).

3.**Define adjacency matrix?**
The adjacency –matrix is an n x n matrix A whose elements aij are given by
aij = 1 if (vi, vj) Exists =0 otherwise

4.**Define adjacent nodes?**

Any two nodes, which are connected by an edge in a graph, are called adjacent nodes. For example, if an edge x E is associated with a pair of nodes

(u,v) where u, v V, then we say that the edge x connects the nodes u and v.

5.**What is a directed graph?**
A graph in which every edge is directed is called a directed graph.

6.**What is an undirected graph?**
A graph in which every edge is undirected is called an undirected graph.

7.**What is a loop?**
An edge of a graph, which connects to itself, is called a loop or sling.

8.**What is a simple graph?**
A simple graph is a graph, which has not more than one edge between a pair of nodes.

9.**What is a weighted graph?**
A graph in which weights are assigned to every edge is called a weighted graph.

10.**Define indegree and out degree of a graph?**
In a directed graph, for any node v, the number of edges, which have v as their initial node, is called the out degree of the node v.

Outdegree: Number of edges having the node v as root node is the outdegree of the node v.

11.Define path in a graph?
The path in a graph is the route taken to reach terminal node from a starting node.

12.**What is a simple path?**
i. A path in a diagram in which the edges are distinct is called a simple path.
ii. It is also called as edge simple.

13.**What is a cycle or a circuit?**

A path which originates and ends in the same node is called a cycle or circuit.

14.**What is an acyclic graph?**

A simple diagram, which does not have any cycles, is called an acyclic graph.

15.**What is meant by strongly connected in a graph?**

An undirected graph is connected, if there is a path from every vertex to every other vertex. A directed graph with this property is called strongly connected.

16.**When a graph said to be weakly connected?**

When a directed graph is not strongly connected but the underlying graph is connected, then the graph is said to be weakly connected.

17.**Name the different ways of representing a graph? Give examples (Nov 10)**

a. Adjacency matrix

b. Adjacency list

18.**What is an undirected acyclic graph?**

When every edge in an acyclic graph is undirected, it is called an undirected acyclic graph. It is also called as undirected forest.

19.**What is meant by depth?**

The depth of a list is the maximum level attributed to any element with in the list or with in any sub list in the list.

20.**What is the use of BFS?**

BFS can be used to find the shortest distance between some starting node and the remaining nodes of the graph. The shortest distance is the minimum number of edges traversed in order to travel from the start node the specific node being examined.

21.**What is topological sort?**

It is an ordering of the vertices in a directed acyclic graph, such that: If there is a path from u to v, then v appears after u in the ordering.

22.Write BFS algorithm

1. **Initialize the first node's dist number and place in queue**

2. Repeat until all nodes have been examined

3. Remove current node to be examined from queue

1. Find all unlabeled nodes adjacent to current node
2. If this is an unvisited node label it and add it to the queue
3. Finished.

23.**Define biconnected graph?**

A graph is called biconnected if there is no single node whose removal causes the graph to break into two or more pieces. A node whose removal causes the graph to become disconnected is called a cut vertex.

24.**What are the two traversal strategies used in traversing a graph?**

a. Breadth first search

b. Depth first search

25.Define Articulation Points (or Cut Vertices) in a Graph

A vertex in an undirected connected graph is an articulation point (or cut vertex) if removing it (and edges through it) disconnects the graph. Articulation points represent vulnerabilities in a connected network – single points whose failure would split the network into 2 or more disconnected components. They are useful for designing reliable networks.

For a disconnected undirected graph, an articulation point is a vertex removing which increases number of connected components.
Following are some example graphs with articulation points encircled with red color.

16 MARKS

1. Explain the various representation of graph with example in detail?
2. Explain Breadth First Search algorithm with example?
3. Explain Depth first and breadth first traversal?
4. What is topological sort? Write an algorithm to perform topological sort?(8) (Nov 09)
5. (i) write an algorithm to determine the biconnected components in the given graph. (10) (may 10)

   (ii)determine the biconnected components in a graph. (6)
6. Explain the various applications of Graphs.

UNIT – V

2 MARKS

1.What is meant by Sorting?

Sorting is ordering of data in an increasing or decreasing fashion according to some linear relationship among the data items.

2. List the different sorting algorithms.

- Bubble sort
- Selection sort
- Insertion sort
- Shell sort
- Quick sort
- Radix sort
- Heap sort
- Merge sort

3. Why bubble sort is called so?

The bubble sort gets its name because as array elements are sorted they gradually **“bubble” to their proper positions, like bubbles risin**g in a glass of soda.

4. State the logic of bubble sort algorithm.

The bubble sort repeatedly compares adjacent elements of an array. The first and second elements are compared and swapped if out of order. Then the second and third elements are compared and swapped if out of order. This sorting process continues until the last two elements of the array are compared and swapped if out of order.

5. What number is always sorted to the top of the list by each pass of the Bubble sort algorithm?

Each pass through the list places the next largest value in its proper place. In essence, each item **“bubbles” up to the location where it belongs.**

6. When does the Bubble Sort Algorithm stop?

The bubble sort stops when it examines the entire array and finds that no "swaps" are needed. The bubble sort keeps track of the occurring swaps by the use of a flag.

7. State the logic of selection sort algorithm.

It finds the lowest value from the collection and moves it to the left. This is repeated until the complete collection is sorted.

8. What is the output of selection sort after the 2nd iteration given the following sequence? 16 3 46 9 28 14

Ans: 3 9 46 16 28 14

9.How does insertion sort algorithm work?

In every iteration an element is compared with all the elements before it. While comparing if it is found that the element can be inserted at a suitable position, then space is created for it by shifting the other elements one position up and inserts the desired element at the suitable position. This procedure is repeated for all the elements in the list until we get the sorted elements.

10.What operation does the insertion sort use to move numbers from the unsorted section to the sorted section of the list?

The Insertion Sort uses the swap operation since it is ordering numbers within a single list.

11. How many key comparisons and assignments an insertion sort makes in its worst case?

The worst case performance in insertion sort occurs when the elements of the input array are in descending order. In that case, the first pass requires one comparison, the second pass **requires two comparisons, third pass three comparisons,….kth pass requires (k**-1), and finally the last pass requires (n-1) comparisons. Therefore, total numbers of comparisons are:

f(n) = 1+2+3+………+(n-k)+…..+(n-2)+(n-1) = n(n-1)/2 = O(n2)

12. Which sorting algorithm is best if the list is already sorted? Why?

Insertion sort as there is no movement of data if the list is already sorted and

complexity is of the order O(N).

13. Which sorting algorithm is easily adaptable to singly linked lists? Why?

Insertion sort is easily adaptable to singly linked list. In this method there is an array link of pointers, one for each of the original array elements. Thus the array can be thought of as a linear link list pointed to by an external pointer first initialized to 0. To insert the k^{th} element the linked list is traversed until the proper position for x[k] is found, or until the end of the list is reached. At that point x[k] can be inserted into the list by merely adjusting the pointers without shifting any elements in the array which reduces insertion time.

14. Why Shell Sort is known diminishing increment sort?

The distance between comparisons decreases as the sorting algorithm runs until the last phase in which adjacent elements are compared. In each step, the sortedness of the sequence is increased, until in the last step it is completely sorted.

15. Which of the following sorting methods would be especially suitable to sort alist L consisting of a sorted list followed by a few "random" elements?

Quick sort is suitable to sort a list L consisting of a sorted list followed by a few "random" elements.

16.What is the output of quick sort after the 3rd iteration given the following sequence?
24 56 47 35 10 90 82 31

Pass 1:- (10) 24 (56 47 35 90 82 31)

Pass 2:- 10 24 (56 47 35 90 82 31)

Pass 3:- 10 24 (47 35 31) 56 (90 82)

17.Mention the different ways to select a pivot element.

The different ways to select a pivot element are

- Pick the first element as pivot
- Pick the last element as pivot
- Pick the Middle element as pivot
- Median-of-three elements
- Pick three elements, and find the median x of these elements
- Use that median as the pivot.
- Randomly pick an element as pivot.

18.What is divide-and-conquer strategy?

- Divide a problem into two or more sub problems
- Solve the sub problems recursively
- Obtain solution to original problem by combining these solutions

19. Compare quick sort and merge sort.

Quicksort has a best-case linear performance when the input is sorted, or nearly sorted. It has a worst-case quadratic performance when the input is sorted in reverse, or nearly sorted in reverse.

Merge sort performance is much more constrained and predictable than the performance of quicksort. The price for that reliability is that the average case of merge sort is slower than the average case of quicksort because the constant factor of merge sort is larger.

20.Define Searching.

Searching for data is one of the fundamental fields of computing. Often, the difference between a fast program and a slow one is the use of a good algorithm for the data set. Naturally, the use of a hash table or binary search tree will result in more efficient searching, but more often than not an array or linked list will be used. It is necessary to understand good ways of searching data structures not designed to support efficient search.

21.What is linear search?

In Linear Search the list is searched sequentially and the position is returned if the key element to be searched is available in the list, otherwise -1 is returned. The search in Linear Search starts at the beginning of an array and move to the end, testing for a match at each item.

22.**What is Binary search?**

A binary search, also called a dichotomizing search, is a digital scheme for locating a specific object in a large set. Each object in the set is given a key. The number of keys is always a power of 2. If there are 32 items in a list, for example, they might be numbered 0 through 31 (binary 00000 through 11111). If there are, say, only 29 items, they can be numbered 0 through 28 (binary 00000 through 11100), with the numbers 29 through31 (binary 11101, 11110, and 11111) as dummy keys.

23.Define hash function?

Hash function takes an identifier and computes the address of that identifier in the hash table using some function.

24.Why do we need a Hash function as a data structure as compared to any other data structure? (may 10)

Hashing is a technique used for performing insertions, deletions, and finds in constant average time.

25.What are the important factors to be considered in designing the hash function? (Nov 10)

- To avoid lot of collision the table size should be prime
- For string data if keys are very long, the hash function will take long to compute.

26.. What do you mean by hash table?

The hash table data structure is merely an array of some fixed size, containing the keys. A key is a string with an associated value. Each key is mapped into some number in the range 0 to tablesize-1 and placed in the appropriate cell.

27. **What do you mean by hash function?**

A hash function is a key to address transformation which acts upon a given key to compute the relative position of the key in an array. The choice of hash function should be simple and it must distribute the data evenly. A simple hash function is hash_key=key mod tablesize.

28.**What do you mean by separate chaining?**

Separate chaining is a collision resolution technique to keep the list of all elements that hash to the same value. This is called separate chaining because each hash table element is a separate chain (linked list). Each linked list contains all the elements whose keys hash to the same index.

16 MARKS

1. Write an algorithm to implement Bubble sort with suitable example.
2. Explain any two techniques to overcome hash collision.
3. Write an algorithm to implement insertion sort with suitable example.
4. Write an algorithm to implement selection sort with suitable example.
5. Write an algorithm to implement radix sort with suitable example.
6. Write an algorithm for binary search with suitable example.
7. Discuss the common collision resolution strategies used in closed hashing system.
8. Given the input { 4371, 1323, 6173, 4199, 4344, 9679, 1989 } and a hash function of h(X)=X (mod 10) show the resulting:
    a. Separate Chaining hash table
    b. Open addressing hash table using linear probing
9. Explain Re-hashing and Extendible hashing.
10. Show the result of inserting the keys 2,3,5,7,11,13,15,6,4 into an initially empty

    extendible hashing data structure with M=3. (8) (Nov 10)
11. what are the advantages and disadvantages of various collision resolution strategies? (6)