CS 8391- DATA STRUCTURES

2 Marks With Answer

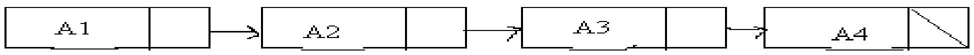
UNIT I – LINEAR STRUCTURES -LIST

1. Define Data Structures

Data Structures is defined as the way of organizing all data items that consider not only the elements stored but also stores the relationship between the elements.

2. Define Linked Lists

Linked list consists of a series of structures, which are not necessarily adjacent in memory. Each structure contains the element and a pointer to a structure containing its successor. We call this theNext Pointer. The last cell'sNext pointer points to NULL.



3. State the different types of linked lists

The different types of linked list include singly linked list, doubly linked list and circular linked list.

4. List the basic operations carried out in a linked list

The basic operations carried out in a linked list include:

• Creation of a list

• Insertion of a node

• Deletion of a node

• Modification of a node

• Traversal of the list

5. List out the advantages of using a linked list

• It is not necessary to specify the number of elements in a linked list during its declaration

- Linked list can grow and shrink in size depending upon the insertion and deletion that occurs in the list
- Insertions and deletions at any place in a list can be handled easily and efficiently
- A linked list does not waste any memory space

6. List out the disadvantages of using a linked list

- Searching a particular element in a list is difficult and time consuming
- A linked list will use more storage space than an array to store the same number of elements

7. List out the applications of a linked list

Some of the important applications of linked lists are manipulation of polynomials, sparse matrices, stacks and queues.

8. State the difference between arrays and linked lists

Arrays	**Linked Lists**
Size of an array is fixed	Size of a list is variable
It is necessary to specify the number of elements during declaration.	It is not necessary to specify the number of elements during declaration
Insertions and deletions are somewhat difficult	Insertions and deletions are carried out easily
It occupies less memory than a linked list for the same number of elements	It occupies more memory

9. Define an Abstract Data Type (ADT)

An abstract data type is a set of operations. ADTs are mathematical abstractions; nowhere in an ADT's definition is there any mention of how the set of operations is implemented. Objects such as lists, sets and graphs, along with their operations can be viewed as abstract data types.

10. What are the advantages of modularity?

- It is much easier to debug small routines than large routines

• It is easier for several people to work on a modular program simultaneously

• A well-written modular program places certain dependencies in only one routine, making changes easier

11. What are the objectives of studying data structures?

• To identify and create useful mathematical entities and operations to determine what classes of problems can be solved using these entities and operations

• To determine the representation of these abstract entities and to implement the abstract operations on these concrete representation

UNIT II - LINEAR DATA STRUCTURES – STACKS, QUEUES

1. What are the types of queues?

• Linear Queues – The queue has two ends, the front end and the rear end. The rear end is where we insert elements and front end is where we delete elements. We can traverse in a linear queue in only one direction ie) from front to rear.

• Circular Queues – Another form of linear queue in which the last position is connected to the first position of the list. The circular queue is similar to linear queue has two ends, the front end and the rear end. The rear end is where we insert elements and front end is where we delete elements. We can traverse in a circular queue in only one direction ie) from front to rear.

• Double-Ended-Queue – Another form of queue in which insertions and
deletions are made at both the front and rear ends of the queue.

2. List the applications of stacks

• Towers of Hanoi

• Reversing a string

• Balanced parenthesis

• Recursion using stack

• Evaluation of arithmetic expressions

3.. List the applications of queues

• Jobs submitted to printer

• Real life line

• Calls to large companies

• Access to limited resources in Universities

• Accessing files from file server

4. Define a stack

Stack is an ordered collection of elements in which insertions and deletions are restricted to one end. The end from which elements are added and/or removed is referred to as top of the stack. Stacks are also referred as piles, push-down lists and last-in-first-out (LIFO) lists.

5. List out the basic operations that can be performed on a stack

The basic operations that can be performed on a stack are

- Push operation
- Pop operation
- Peek operation
- Empty check
- Fully occupied check

6. State the different ways of representing expressions

The different ways of representing expressions are

- Infix Notation
- Prefix Notation
- Postfix Notation

7. State the rules to be followed during infix to postfix conversions

- Fully parenthesize the expression starting from left to right. During parenthesizing, the operators having higher precedence are first parenthesized
- Move the operators one by one to their right, such that each operator replaces their corresponding right parenthesis
- The part of the expression, which has been converted into postfix is to be treated as single operand

8. Mention the advantages of representing stacks using linked lists than arrays

- It is not necessary to specify the number of elements to be stored in a stack during its declaration, since memory is allocated dynamically at run time when an element is added to the stack
- Insertions and deletions can be handled easily and efficiently
- Linked list representation of stacks can grow and shrink in size without wasting memory space, depending upon the insertion and deletion that occurs in the list
- Multiple stacks can be represented efficiently using a chain for each stack

9. Mention the advantages of representing stacks using linked lists than arrays

- It is not necessary to specify the number of elements to be stored in a stack during its declaration, since memory is allocated dynamically at run time when an element is added to the stack
- Insertions and deletions can be handled easily and efficiently
- Linked list representation of stacks can grow and shrink in size without wasting memory space, depending upon the insertion and deletion that occurs in the list
- Multiple stacks can be represented efficiently using a chain for each stack

10. Define a queue

Queue is an ordered collection of elements in which insertions are restricted to one end called the rear end and deletions are restricted to other end called the front end. Queues are also referred as First-In-First-Out (FIFO) Lists.

11. Define a priority queue

Priority queue is a collection of elements, each containing a key referred as the priority for that element. Elements can be inserted in any order (i.e., of alternating priority), but are arranged in order of their priority value in the queue. The elements are deleted from the queue in the order of their priority (i.e., the elements with the highest priority is deleted first). The elements with the same priority are given equal importance and processed accordingly.

12. State the difference between queues and linked lists

The difference between queues and linked lists is that insertions and deletions may occur anywhere in the linked list, but in queues insertions can be made only in the rear end and deletions can be made only in the front end.

13.. Define a Deque

Deque (Double-Ended Queue) is another form of a queue in which insertions and deletions are made at both the front and rear ends of the queue. There are two variations of a deque, namely, input restricted deque and output restricted deque. The input
restricted deque allows insertion at one end (it can be either front or rear) only. The
output restricted deque allows deletion at one end (it can be either front or rear)

14. What is the need for Priority queue?

In a multiuser environment, the operating system scheduler must decide which of the several processes to run only for a fixed period of time. One algorithm uses queue. Jobs are initially placed at the end of the queue. The scheduler will repeatedly take the first job on the queue, run it until either it finishes or its time limit is up and place it at the end of the queue if it does not finish. This strategy is not appropriate, because very short jobs will soon to take a long time because of the wait involved in the run.

Generally, it is important that short jobs finish as fast as possible, so these jobs should have precedence over jobs that have already been running. Further more, some jobs that are not short are still very important and should have precedence. This particular application seems to require a special kind of queue, known as priority queue. Priority queue is also called as Heap or Binary Heap.

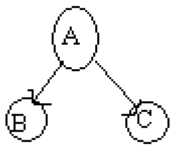
UNIT III – NON LINEAR DATA STRUCTURES – TREES

1. Define a tree

A tree is a collection of nodes. The collection can be empty; otherwise, a tree consists of a distinguished node r, called the root, and zero or more nonempty (sub) trees T1, T2,…,Tk, each of whose roots are connected by a directed edge from r.

2. Define root

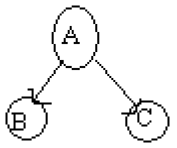
This is the unique node in the tree to which further sub-trees are attached.



Here, A is the root.

3. Define degree of the node

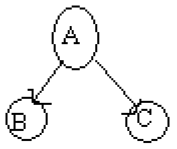
The total number of sub-trees attached to that node is called the degree of the node.



For node A, the degree is 2 and for B and C, the degree is 0.

4. Define leaves

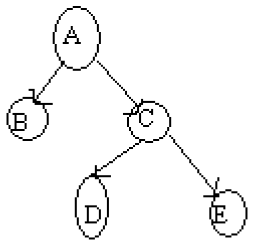
These are the terminal nodes of the tree. The nodes with degree 0 are always the leaves.



Here, B and C are leaf nodes.

5. Define internal nodes

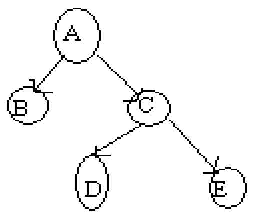
The nodes other than the root and the leaves are called internal nodes.



Here, C is the internal node.

6. Define parent node

The node which is having further sub-branches is called the parent node of those sub-branches.



Here, node C is the parent node of D and E

7. Define depth and height of a node

For any node ni, the depth of ni is the length of the unique path from the root to ni. The height of ni is the length of the longest path from ni to a leaf.

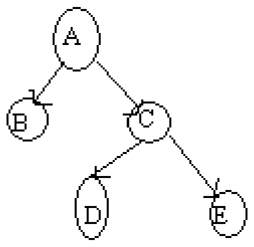
8. Define depth and height of a tree

The depth of the tree is the depth of the deepest leaf. The height of the tree is equal to the height of the root. Always depth of the tree is equal to height of the tree.

9. What do you mean by level of the tree?

The root node is always considered at level zero, then its adjacent children are supposed to be at level 1 and so on.

Here, node A is at level 0, nodes B and C are at level 1 and nodes D and E are at level 2.



10. Define forest

A tree may be defined as a forest in which only a single node (root) has no predecessors. Any forest consists of a collection of trees.

11. Define a binary tree

A binary tree is a finite set of nodes which is either empty or consists of a root and two disjoint binary trees called the left sub-tree and right sub-tree.

12. Define a path in a tree

A path in a tree is a sequence of distinct nodes in which successive nodes are connected by edges in the tree.

13. Define a full binary tree

A full binary tree is a tree in which all the leaves are on the same level and every non-leaf node has exactly two children.

14. Define a complete binary tree

A complete binary tree is a tree in which every non-leaf node has exactly two children not necessarily to be on the same level.

15. State the properties of a binary tree

- The maximum number of nodes on level n of a binary tree is 2n-1, where n≥1.
- The maximum number of nodes in a binary tree of height n is 2n-1, where n≥1.
- For any non-empty tree, nl=nd+1 where nl is the number of leaf nodes and nd is the number of nodes of degree 2.

16. What is meant by binary tree traversal?

Traversing a binary tree means moving through all the nodes in the binary tree, visiting each node in the tree only once.

17. What are the different binary tree traversal techniques?

- Preorder traversal
- Inorder traversal
- Postorder traversal
- Levelorder traversal

18. What are the tasks performed during inorder traversal?

- Traverse the left sub-tree
- Process the root node
- Traverse the right sub-tree

19. What are the tasks performed during postorder traversal?

- Traverse the left sub-tree
- Traverse the right sub-tree
- Process the root node

20. State the merits of linear representation of binary trees.

- Storage method is easy and can be easily implemented in arrays
- When the location of a parent/child node is known, other one can be determined easily
- It requires static memory allocation so it is easily implemented in all programming language

21. State the demerit of linear representation of binary trees.

Insertions and deletions in a node take an excessive amount of processing time due to data movement up and down the array.

22. State the merit of linked representation of binary trees.

Insertions and deletions in a node involve no data movement except the rearrangement of pointers, hence less processing time.

23. State the demerits of linked representation of binary trees.

• Given a node structure, it is difficult to determine its parent node

• Memory spaces are wasted for storing null pointers for the nodes, which have one or no sub-trees

• It requires dynamic memory allocation, which is not possible in some programming language

24. Define a binary search tree

A binary search tree is a special binary tree, which is either empty or it should satisfy the following characteristics:

Every node has a value and no two nodes should have the same value i.e) the values in the binary search tree are distinct

• The values in any left sub-tree is less than the value of its parent node

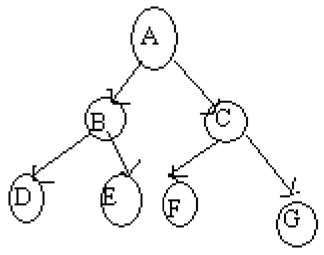
• The values in any right sub-tree is greater than the value of its parent node

• The left and right sub-trees of each node are again binary search trees

25. What is the use of threaded binary tree?

In threaded binary tree, the NULL pointers are replaced by some addresses. The left pointer of the node points to its predecessor and the right pointer of the node points to its successor.

26. Traverse the given tree using Inorder, Preorder and Postorder traversals.

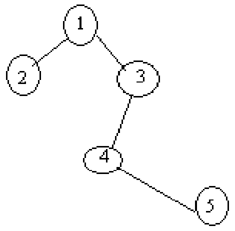


Inorder : D H B E A F C I G J

Preorder: A B D H E C F G I J

Postorder: H D E B F I J G C A

27. In the given binary tree, using array you can store the node 4 at which location?



At location 6

1	2	3	-	-	4	-	-	5

where LCn means Left Child of node n and RCn means Right Child of node n

28. Define AVL Tree.

An empty tree is height balanced. If T is a non-empty binary tree with TL and TR as its left and right subtrees, then T is height balanced if

i) TL and TR are height balanced and

ii) $|hL - hR| \leq 1$

Where hL and hR are the heights of TL and TR respectively.

29. What do you mean by balanced trees?

Balanced trees have the structure of binary trees and obey binary search tree properties. Apart from these properties, they have some special constraints, which differ from one data structure to another. However, these constraints are aimed only at reducing the height of the tree, because this factor determines the time complexity.

Eg: AVL trees, Splay trees.

30.What are the categories of AVL rotations?

Let A be the nearest ancestor of the newly inserted nod which has the balancing factor ±2. Then the rotations can be classified into the following four categories:

Left-Left: The newly inserted node is in the left subtree of the left child of A.

Right-Right: The newly inserted node is in the right subtree of the right child of

A. Left-Right: The newly inserted node is in the right subtree of the left child of A.

Right-Left: The newly inserted node is in the left subtree of the right child of A.

31.What do you mean by balance factor of a node in AVL tree?

The height of left subtree minus height of right subtree is called balance factor of a node in AVL tree.The balance factor may be either 0 or +1 or -1.The height of an empty tree is -1.

32. Define splay tree.

A splay tree is a binary search tree in which restructuring is done using a scheme called splay. The splay is a heuristic method which moves a given vertex v to the root of the splay tree using a sequence of rotations.

33. What is the idea behind splaying?

Splaying reduces the total accessing time if the most frequently accessed node is moved towards the root. It does not require to maintain any information regarding the height or balance factor and hence saves space and simplifies the code to some extent.

UNIT IV –GRAPHS NON LINEAR DATA STRUCTUURES - GRAPHS

1. 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 for edge E to a set of pairs of elements of V. It can also be represented as G=(V, E).

2. 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 ε E is associated with a pair of nodes (u,v) where u, v ε V, then we say that the edge x connects the nodes u and v.

3. What is a directed graph?

A graph in which every edge is directed is called a directed graph.

4. What is an undirected graph?

A graph in which every edge is undirected is called a directed graph.

5. What is a loop?

An edge of a graph which connects to itself is called a loop or sling.

6. What is a simple graph?

A simple graph is a graph, which has not more than one edge between a pair of nodes than such a graph is called a simple graph.

7. What is a weighted graph?

A graph in which weights are assigned to every edge is called a weighted graph.

8. Define outdegree 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.

9. Define indegree of a graph?

In a directed graph, for any node v, the number of edges which have v as their terminal node is called the indegree of the node v.

10. Define path in a graph?

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

11. What is a simple path?

A path in a diagram in which the edges are distinct is called a simple path. It is also called as edge simple.

12. What is a cycle or a circuit?

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

13. What is an acyclic graph?

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

14. 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.

15. When is 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.

16.Namethe different ways of representing a graph?

a.Adjacencymatrix

b. Adjacency list

17. 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.

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

a.Breadthfirstsearch

b. Depth first search

19. What is a minimum spanning tree?

A minimum spanning tree of an undirected graph G is a tree formed from graph edges that connects all the vertices of G at the lowest total cost.

20. Name two algorithms two find minimum spanning tree

Kruskal'salgorithm

Prim's algorithm

21. Define graph traversals.

Traversing a graph is an efficient way to visit each vertex and edge exactly once.

22. List the two important key points of depth first search.

i) If path exists from one node to another node, walk across the edge – exploring the edge.

ii) If path does not exist from one specific node to any other node, return to the previous node where we have been before – backtracking.

23. What do you mean by breadth first search (BFS)?

BFS performs simultaneous explorations starting from a common point and spreading out independently.

24.DifferentiateBFSandDFS.

No.	DFS	BFS
1.	Backtracking is possible from a dead end	Backtracking is not possible
2.	Vertices from which exploration is incomplete are processed in a	The vertices to be explored are organized as a
3.	Search is done in one particular direction	The vertices in the same level are maintained

25. What do you mean by tree edge?

If w is undiscovered at the time vw is explored, then vw is called a tree edge and v becomes the parent of w.

26. What do you mean by back edge?

If w is the ancestor of v, then vw is called a back edge.

27. Define biconnectivity.

A connected graph G is said to be biconnected, if it remains connected after removal of any one vertex and the edges that are incident upon that vertex. A connected graph is biconnected, if it has no articulation points.

28. What do you mean by articulation point?

If a graph is not biconnected, the vertices whose removal would disconnect the graph are known as articulation points.

29. What do you mean by shortest path?

A path having minimum weight between two vertices is known as shortest path, in which weight is always a positive number.

30. Define Activity node graph.

Activity node graphs represent a set of activities and scheduling constraints. Each node represents an activity (task), and an edge represents the next activity.

31. Define adjacency list.

Adjacency list is an array indexed by vertex number containing linked lists. Each node Vi the ith array entry contains a list with information on all edges of G that leave Vi. It is used to represent the graph related problems.

UNIT V – HASHING AND SET SEARCHING, SORTING AND HASHING TECHNIQUES

1. **Define sorting**

   Sorting arranges the numerical and alphabetical data present in a list in a specific order or sequence. There are a number of sorting techniques available. The algorithms can be chosen based on the following factors

   – Size of the data structure
   – Algorithm efficiency
   – Programmer's knowledge of the technique.

2. **Mention the types of sorting**

   - Internal sorting
   - External sorting

3. **What do you mean by internal and external sorting?**

   An **internal sort** is any data sorting process that takes place entirely within the main memory of a computer. This is possible whenever the data to be sorted is small enough to all be held in the main memory.

   External sorting is a term for a class of sorting algorithms that can handle massive amounts of data. External sorting is required when the data being sorted do not fit into the main memory of a computing device (usually RAM) and instead they must reside in the slower external memory (usually a hard drive)

4. **Define bubble sort**

   Bubble sort is a simple sorting algorithm that works by repeatedly stepping through the list to be sorted, comparing each pair of adjacent items and swapping them if they are in the wrong order. The pass through the list is repeated until no swaps are needed, which indicates that the list is sorted. The algorithm gets its name from the way smaller elements "bubble" to the top of the list.

5. **How the insertion sort is done with the array?**

   It sorts a list of elements by inserting each successive element in the previously sorted sublist.

   Consider an array to be sorted A[1],A[2],….A[n]

   a. Pass 1 : A[2] is compared with A[1] and placed them in sorted order.
   b. Pass 2 : A[3] is compared with both A[1] and A[2] and inserted at an appropriate place. This makes A[1], A[2],A[3] as a sorted sub array.
   c. Pass n-1 : A[n] is compared with each element in the sub array A[1],A[2],……A[n-1] and inserted at an appropriate position.

6. **What are the steps for selection sort?**

   - The algorithm divides the input list into two parts: the sublist of items already sorted, which is built up from left to right at the front (left) of the list, and the sublist of items remaining to be sorted that occupy the rest of the list.
   - Initially, the sorted sublist is empty and the unsorted sublist is the entire input list.
   - The algorithm proceeds by finding the smallest (or largest, depending on sorting order) element in the unsorted sublist, exchanging it with the leftmost unsorted element (putting it in sorted order), and moving the sublist boundaries one element to the right.

7. **What is meant by shell sort?**

   Shell sort, also known as **Shell sort** or **Shell's method**, is an in-place comparison sort. It can either be seen as a generalization of sorting by exchange (bubble sort) or sorting by insertion (insertion sort).[1] The method starts by sorting elements far apart from each other and progressively reducing the gap between them. Starting with far apart elements can move some out-of-place elements into position faster than a simple nearest neighbor exchange. Donald Shell published the first version of this sort in 1959. The running time of Shell sort is heavily dependent on the gap sequence it uses

8. **What are the steps in quick sort?**

The steps are:

a. Pick an element, called a **pivot**, from the list.
b. Reorder the list so that all elements with values less than the pivot come before the pivot, while all elements with values greater than the pivot come after it (equal values can go either way). After this partitioning, the pivot is in its final position. This is called the **partition** operation.
c. Recursively apply the above steps to the sub-list of elements with smaller values and separately to the sub-list of elements with greater values.

9. Define radix sort

Radix Sort is a clever and intuitive little sorting algorithm. **Radix sort** is a non-comparative integer sorting algorithm that sorts data with integer keys by grouping keys by the individual digits which share the same significant position and value. Radix Sort puts the elements in order by comparing the **digits of the numbers**.

10. What are the advantages of insertion sort

Advantages

a. Simplest sorting technique and easy to implement
b. It performs well in the case of smaller lists.
c. It leverages the presence of any existing sort pattern in the list

Disadvantages

- Efficiency of O(n) is not well suited for large sized lists
- It requires large number of elements to be shifted

11. Define searching

Searching refers to determining whether an element is present in a given list of elements or not. If the element is present, the search is considered as successful, otherwise it is considered as an unsuccessful search. The choice of a searching technique is based on the following factors

a. Order of elements in the list i.e., random or sorted
b. Size of the list

12. Mention the types of searching

The types are

- Linear search
- Binary search

13. What is meant by linear search?

Linear search or **sequential search** is a method for finding a particular value in a list that consists of checking every one of its elements, one at a time and in sequence, until the desired one is found.

14. What is binary search?

For binary search, the array should be arranged in ascending or descending order.
In each step, the algorithm compares the search key value with the middle element of the array. If the key match, then a matching element has been found and its index, or position, is returned.
Otherwise, if the search key is less than the middle element, then the algorithm repeats its action on the sub-array to the left of the middle element or, if the search key is greater, on the sub-array to the right.

15. Define hashing function

A hashing function is a key-to-transformation, which acts upon a given key to compute the relative position of the key in an array.
A simple hash function
HASH(KEY_Value)= (KEY_Value)mod(Table-size)

16. What is open addressing?

Open addressing is also called closed hashing, which is an alternative to resolve the collisions with linked lists. In this hashing system, if a collision occurs, alternative cells are tired until an empty cell is found.

There are three strategies in open addressing:

- Linear probing
- Quadratic probing
- Double hashing

17. What are the collision resolution methods?

The following are the collision resolution methods

- Separate chaining
- Open addressing
- Multiple hashing

18. Define separate chaining

It is an open hashing technique. A pointer field is added to each record location, when an overflow occurs, this pointer is set to point to overflow blocks making a linked list. In this method, the table can never overflow, since the linked lists are only extended upon the arrival of new keys

19. . Define Hashing.

Hashing is the transformation of string of characters into a usually shorter fixed length value or key that represents the original string. Hashing is used to index and retrieve items in a database because it is faster to find the item using the short hashed key than to find it using the original value.

20. 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.

21. . 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.

22. . Write the importance of hashing.

- Maps key with the corresponding value using hash function.
- Hash tables support the efficient addition of new entries and the time spent on searching for the required data is independent of the number of items stored.

23. . What do you mean by collision in hashing?

When an element is inserted, it hashes to the same value as an already inserted element, and then it produces collision.

24. What are the collision resolution methods?

- Separate chaining or External hashing
- Open addressing or Closed hashing

25. . 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.

26. . Write the advantage of separate chaining.

- More number of elements can be inserted as it uses linked lists.

27. . Write the disadvantages of separate chaining.

- The elements are evenly distributed. Some elements may have more elements and some may not have anything.
- It requires pointers. This leads to slow the algorithm down a bit because of the time required to allocate new cells, and also essentially requires the implementation of a second data structure.

28. . What do you mean by open addressing?

Open addressing is a collision resolving strategy in which, if collision occurs alternative cells are tried until an empty cell is found. The cells h0(x), h1(x), h2(x),…. are tried in succession, where hi(x)=(Hash(x)+F(i))mod Tablesize with F(0)=0. The function F is the collision resolution strategy.

29. What are the types of collision resolution strategies in open addressing?

- Linear probing
- Quadratic probing
- Double hashing

30. What do you mean by Probing?

Probing is the process of getting next available hash table array cell.

31. . What do you mean by linear probing?

Linear probing is an open addressing collision resolution strategy in which F is a linear function of i, F(i)=i. This amounts to trying sequentially in search of an empty cell. If the table is big enough, a free cell can always be found, but the time to do so can get quite large.

32. What do you mean by primary clustering?

In linear probing collision resolution strategy, even if the table is relatively empty, blocks of occupied cells start forming. This effect is known as primary clustering means that any key hashes into the cluster will require several attempts to resolve the collision and then it will add to the cluster.

33. What do you mean by quadratic probing?

Quadratic probing is an open addressing collision resolution strategy in which F(i)=i2. There is no guarantee of finding an empty cell once the table gets half full if the table size is not prime. This is because at most half of the table can be used as alternative locations to resolve collisions.

34. What do you mean by secondary clustering?

Although quadratic probing eliminates primary clustering, elements that hash to the same position will probe the same alternative cells. This is known as secondary clustering.

35. What do you mean by double hashing?

Double hashing is an open addressing collision resolution strategy in which F(i)=i.hash2(X). This formula says that we apply a second hash function to X and probe at a distance hash2(X), 2hash2(X),….,and so on. A function such as hash2(X)=R-(XmodR), with R a prime smaller than

36. What do you mean by rehashing?

If the table gets too full, the running time for the operations will start taking too long and inserts might fail for open addressing with quadratic resolution. A solution to this is to build another table that is about twice as big with the associated new hash function and scan down the entire original hash table, computing the new hash value for each element and inserting it in the new table. This entire operation is called rehashing.

37. What is the need for extendible hashing?

If either open addressing hashing or separate chaining hashing is used, the major problem is that collisions could cause several blocks to be examined during a Find, even for a well-distributed hash table. Extendible hashing allows a find to be performed in two disk accesses. Insertions also require few disk accesses.

38. List the limitations of linear probing.

- Time taken for finding the next available cell is large.
- In linear probing, we come across a problem known as clustering.

39. Mention one advantage and disadvantage of using quadratic probing.

Advantage: The problem of primary clustering is eliminated.

Disadvantage: There is no guarantee of finding an unoccupied cell once the table is nearly half full.

DATA STRUCTURES VIVA QUESTIONS

1Q) What is a Data Structure?

Ans) A **Data Structure** is a data object together with the relationships that exists among the instances & among the individual elements that compose an instance.

2Q) Types of Data Structures and give examples?

Ans) There are two types of Data Structures:

1. **Linear Data Structures:** A data structure is said to be linear if the elements form a sequence. It is sequential and continues in nature i.e. access the data in sequential manner.

   In linear data structure we can not insert an item in middle place and it maintains a linear relationship between its elements

   egs: Array, Linked list, Stack, Queue, Dequeue etc.

2. **Non Linear Data Structures:** A data structure is said to be non-linear if elements do not form a sequence. (Not sequential).

   It does not maintain any linear relationship between their elements. Every data item is attached to several other data items in a way that is specific for reflecting relationships. The data items are not arranged in a sequential structure.

   egs: Trees, Graphs.

[A data structure is linear if every item is related with next and previous item and it is non linear if it is attach with many of the items in specific ways to reflect relationship.]

3Q) What is a Singly Linked List?

Ans) Singly Linked List is a Sequence of dynamically allocated Storage elements, each element of which contains a pointer to its successor. A pointer to the first element of the list is called as head and a pointer to the last element of the list is called as tail used to keep track of the list elements.

4Q) What is Doubly Linked List?

Ans) In Doubly Linked List each element contains two pointers: One Pointer points to its successor and another to its predecessor (previous element).It is also called as two way linked list (traversing can be done in both directions).

5Q) Differentiate Array and Linked List?

Ans)

Array	Linked List
1.Size of the array is fixed	1.Size of the linked list is not fixed
2. Memory is allocated Statically (or) Dynamically (at run time). (If the memory is allocated for an array Statically(at compile time) it is called Static Array and if memory is allocated at run time (dynamically)using operator new it is called Dynamic Array)	2. Memory is allocated dynamically (at runtime).
3.Memory wastage will be there if all the array positions are not utilized	3.Memory is not wasted as only Required memory is allocated

STACKS: (LIFO DATA STRUCTURE)

6Q) What is a Stack? (LIFO Data Structure)

Ans) Stack is an ordered collection of items into which items can be inserted and deleted from only one end called as "Top" of the Stack. It is also called as LIFO list.(Last In First Out).

7Q) What is Stack Underflow?

Ans) Is Stack is empty and POP operation is performed it is not possible to delete the items. This situation is called Stack Underflow.

8Q) What is Stack Overflow?

Ans) If Stack is full and PUSH operation is performed it is not possible to insert or Push the new items into the stack. This situation is called Stack Overflow.

9Q) What are the Applications of Stack?

Ans) i) Stacks are used to convert Infix expression into Postfix.

ii) Stacks are used to Evaluate Postfix Expression.

iii) Stacks are used in recursion etc.

10Q) What is the use of Postfix expressions?

Ans) Postfix Expressions are easy to evaluate as postfix expressions does not make use of operator precedence not does it require the use of parenthesis.

QUEUES: (FIFO DATA STRUCTURE)

11Q) What is a Queue?

Ans) It is an ordered collection of items into which items can be inserted from one end called as REAR end and items are deleted from other end called as FRONT end of the Queue. It is also called as FIRST IN FIRST OUT (FIFO) LIST).

12Q) What are the applications of Queues?

Ans) i) Queues are used in Breadth First Traversal of a Tree.

ii) Queues are used in implementation of Scheduling algorithms of Operating Systems.

13Q) What is a Circular Queue?

Ans) In Circular Queue, the first position of the array is kept behind the last position of the array.

14Q) Differentiate Linear Queue and Circular Queue?

Ans) In Linear Queue once the queue is full and the deletion is performed, even if first position is free(vacant) it is not possible to insert the item in that position whereas in Circular Queue it is possible since the first position is kept behind the last position.

15Q) What is Dequeue? (Double Ended Queue)

Ans) In Double Ended Queue insertion and deletion are possible from both the ends.

TREES:

16Q) What is a Tree?

Ans) Tree is a finite non-empty set of nodes with the following properties:

i) A designated node of the set is called as root of the tree and
ii) The remaining nodes are partitioned into n>=0 subsets, each of which is a tree.

Degree of a node: The number of sub trees attached to a node is called degree of that node and the maximum degree of any node in a tree is called **degree of that tree.**

[Note: In a general tree degree of a node is not fixed]

Nodes that have degree zero are called **Leaf or Terminal Nodes**. Consequently, the other nodes are referred to as **Non-Terminals.**

The **Level of a node** is defined by letting the root be at level o or 1.

The **height or depth of a tree** is defined to be the maximum level of any node in the tree.

17Q) What is a Binary Tree?

Ans) A Binary tree T is a finite set of nodes with the following properties:

i) Either the set is empty, T=Ø or
ii) The set consists of a root and exactly two distinct binary trees TL and TR,T={r,TL,TR}.TL is the left subtree and TR is the right subtree of T.

[Note: Maximum degree of any node in a binary tree is 2.Degree of a node in a Binary Tree be either 0 or 1 or 2]

18Q) What is Tree Traversal? List different Tree Traversal Techniques?

Ans) Visiting all nodes of a tree exactly once in a systematic way is called Tree Traversal. Different types of tree traversals are

i) **Depth First Traversals**: PREOREDER (N L R) ,INORDER (L N R) & POSTORDER (L R N) Traversals.
ii) **Breadth First Traversal** (or) **Level Order Traversal** (Visiting Level by level from left to right)

19Q) What is a Binary Search Tree? Give one example?

Ans) A Binary Search Tree T is a finite set of keys. Either the set is empty T=Ø ,or the set consists of a root "r" and exactly two binary search trees TL and TR,T = {r,TL,TR} with the following properties:

i) All the keys contained in the left subtree are less than the root key.
ii) All the keys contained in the right subtree are larger than the root key.

[Note: Duplicates are not allowed in a Binary Search Tree]

20Q) What is the best, average and worst case time complexity of insertion, deletion and Search operations in a Binary Search Tree?

Ans) In Best and Avg case **O(log n)** and in Worst case **O(n).**

21Q) What is an AVL Search Tree? What is AVL Balance Condition?

Ans) An AVL Search Tree is a balanced binary search tree. An empty binary tree is AVL balanced. A non – empty binary tree, T={r,TL,TR} is AVL balanced if both TL & TR are AVL balanced and $| h_L - h_R | <=1$,

Where : h_L is the Height of the left subtree and h_R is the Height of the right subtree.

[**Note** : Allowable balance factors of any node is an AVL tree are 0 or 1 or -1 and the use of balancing a binary search tree is even in worst case the time complexity of insert, delete and search operations is reduced to **O(log n)** from **O(n)**]

22Q) What is a Full (perfect) Binary Tree?

Ans) If a binary tree of height 'h' has exactly **[2^{h+1} -1]** nodes then that binary tree is called as Full Binary Tree.

23Q) What is a Complete Binary Tree?

Ans) Complete Binary Tree is a binary tree T = {r,TL,TR} with the following properties:

If "i" is the index of any node in a complete binary tree then:

i) The parent of "i" is at position "i/2" [if i=1 it is the root node and has no parent]
ii) The left child of node "i" is at position "2i" [if 2i>n then no left child exists]
iii) The right child of node "i" is at position "2i+1" [if 2i+1>n then no right child exists]

[Note: In a complete binary tree nodes are filled level by level from left to right]

24Q) List one application of trees (Search trees)?

Ans) Search trees are used to implement dictionaries.

PRIORITY QUEUES:

25Q) What is Priority Queue and differentiate Priority Queue and Queue?

Ans) In Priority Queue each item is associated with a priority.

In Priority Queue items can be inserted arbitrary order but the items are deleted based upon the priority that is the item with highest priority is deleted first. Whereas in Queue the items are inserted from rear end and deleted from front end and the item which is inserted first is deleted first (FIFO).

26Q) What is a Min Heap?

Ans) Min Heap is a Complete Binary Tree in which the key value at parent is always less than or equal to its child values.

27Q) What is a Max Heap?

Ans) In Max Heap the key value at parent is always larger or equal to its child values.

28Q) What is a (Binary) Heap?

Ans) A (Binary) Heap is a Complete Binary Tree with Heap Ordered Property (Min Heap or Max Heap) **[Note: Duplicates are allowed in a Heap]**

GRAPHS:

29Q) What is a Graph? Name various Graph Representation Techniques?

Ans) A Graph G = (V,E) is Set of Vertices and Edges.

A Graph can be represented as an Adjacency Matrix (or) as an Adjacency List.

30Q) What is difference between a Graph and a Tree?

Ans) A Graph contains Cycles (loops) but a Tree does not contain any cycles.

[A Tree contains a root node but Graph does not contain the root node.]

31Q) What is a Spanning Tree?

Ans) A Spanning Tree T = (V', E') is a Sub Graph of G = (V,E) with following properties:

i) V = V' [The vertices in a graph and spanning tree are same]
ii) T is Acyclic.
iii) T is connected.

[Note: I f Graph has “n” vertices the Spanning Tree contains exactly “n - 1” edges]

32Q) Name the methods to construct a Minimum cost Spanning Tree?

Ans) Prim’s method and Kruskal’s method.

SORTING & SEARCHING:

33Q) What is Linear Search? What is the time complexity of searching an item in a list using linear search?

Ans) In linear search the item to be searched is compared with each item starting with the item in the first position of the array until the item is found or all the items of the array. The time complexity of linear search in Worst case is **O(n).**

34Q) What is Binary Search method? What is the time complexity?

Ans) Binary Search method can be used only if the list is Sorted. In binary search method first the array is divided in to two by finding the mi position of the array and the item to be searched is compared with the mid value.

- If the item to be searched is less than the mid value, the item is searched in the left subarray.
- If the item is larger than the mid value it is searched in the right sub array recursively.
- Otherwise it equals to the mid value and search is successful.
- The time complexity of Binary Search method is **O(log n).**

35Q) What is Sorting?

Ans) Arranging the elements in (ascending or descending) some particular order is called Sorting.

36Q) Study the procedure of following Sorting techniques?

Ans) i) MERGE SORT ii) QUICK SORT iii) INSERTION SORT iv) SELECTION SORT v) HEAP SORT

37Q) Time complexities of Sorting Techniques:

SORTING TECHNIQUE	BEST CASE	AVG CASE	WORST CASE
INSERTION SORT	**O(n)**	**O(n^2)**	**O(n^2)**
SELECTION SORT	**O(n^2)**	**O(n^2)**	**O(n^2)**
MERGE SORT	**O(n log n)**	**O(n log n)**	**O(n log n)**
QUICK SORT	**O(n log n)**	**O(n log n)**	**O(n^2)**
HEAP SORT	**O(n log n)**	**O(n log n)**	**O(n log n)**

[NOTE: If the given array is already sorted then Insertion Sort technique is best to sort the given sequence.]

<u>Study the procedures of all the sorting methods.</u>

38Q) Comparison between Quick Sort and Heap Sort?

Ans) If the size of the array to be sorted is very large the Heap sort is efficient than Quick sort since in worst case the time complexity of Heap sort remains same as O (n log n) but Quick Sort time complexity changes to O (n^2).But if the size of the Array is small then Quick sort is efficient than Heap sort.

HASHING:

39Q) What is Hashing?

Ans) Hashing is used to determine the position of a Key by using the Key itself. To determine the position of the key it makes use of Function called Hash function.

A **hash function** takes a group of characters (called a key) and maps it to a value of a certain length (called a hash value or hash). The hash value is representative of the original string of characters, but is normally smaller than the original.

(or)

A **hash function** maps a key into a bucket in the hash table.

40Q) List different Hashing methods.

Ans) i) Division method ii) Mid Square method iii) Folding method iv) Digit Analysis method

- **Hashing Methods**
  There are eight hashing methods they are:

1. **Modulo-division**
2. **Midsquare**
3. **Digit Extraction (or) Digit Analysis**
4. **Folding**

1. **Modulo-division Method:** This is also known as Division Remainder method.
   This Hash function assumes the keys are non-negative integers.The home bucket is obtaind by using the modulo(%) operator.
   The key is divided by some number D and the remainder is used as the home bucket for k.
   h(k) = k % D
   This function gives bucket addresses in the range 0 through D-1,so the hash table must have at least b = D buckets.

(or)

- This algorithm works with any list size, but a list size that is a prime number produces fewer collisions than other list sizes.
- The formula to calculate the address is:
  Address = key MODULO listsize + 1
  Where listsize is the number of elements in the arrary.
  Example:
  Given data : Keys are : 137456 214562 140145
  137456 % 19 +1 = 11, 214562 % 19 + 1 = 15, 140145 % 19 + 1 = 2

2. **Midsquare Method**
   The Mid – square hash function determines the home bucket for a key by squaring the key and then using an appropriate number of bits from the middle of the square to obtain the bucket address; the key is assumed to be an integer.

   (or)

- In midsquare hashing the key is squared and the address is selected from the middle of the square number.
- Limitation is the size of the key.
  Example:
  $9452^2 = 89340304$: address is 3403

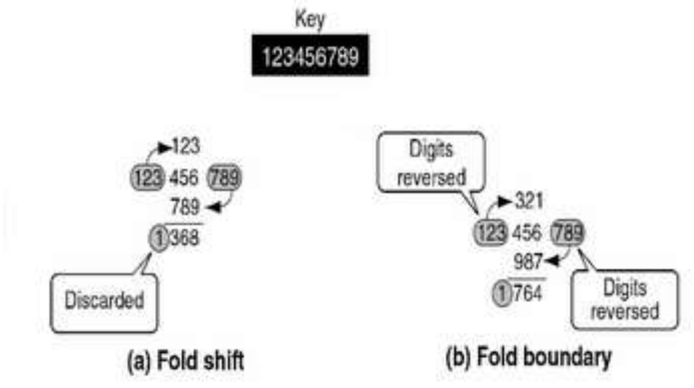
3. **Folding Method**
   Two folding methods are used they are:

i) Fold shift
ii) Fold boundary

   i) Fold Shift
   In fold shift the key value is divided into parts whose size matches the size of the required address. Then the left and right parts are shifted and added with the middle part.

   ii) Fold boundary
   In fold boundary the left and right numbers are folded on a fixed boundary between them and the center number. The two outside values are thus reversed.

   Example:



(a) Fold shift (b) Fold boundary

4. **Digit-extraction (or) Digit Analysis Method:** Using digit extraction selected digits are extracted from the key and used as the address.

   Example: Using six-digit employee number to hash to a three digit address (000-999), we could select the first, third, and fourth digits(from the left) and use them as the address.

   The keys are: 379452 -> 394, 121267 -> 112, 378845 -> 388

41Q) What is Collision?

Ans) By using the hash function if two different keys are hashed to the same position, this situation is called Collision. That is: If x and y are two different ($x \neq y$) and $h(x) = h(y)$ this situation is called Collision.

- A collision occurs when the home bucket for a new pair is occupied by a pair with a different key.
- An overflow occurs when there is no space in the home bucket for the new pair.
- When a bucket can hold only one pair, collisions and overflows occur together.
- Need a method to handle overflows.

Uniform Hash Function: A uniform hash function maps the keys in keySpace into buckets such that approximately the same number of keys get mapped into each bucket.

OVERFLOW HANDLING (OR) COLLISION RESOLUTION TECHNIQUES

- An overflow occurs when the home bucket for a new pair (key, element) is full.
- We may handle overflows by:
  - **Open Addressing:** Search the hash table in some systematic fashion for a bucket that is not full.
    - Linear probing (linear open addressing).
    - Quadratic probing.
    - Rehashing.
    - Random probing.
  - **Chaining:** Eliminate overflows by permitting each bucket to keep a list of all pairs for which it is the home bucket.
    - Array linear list.
    - Chain.

42Q) List different collision resolution strategies?

Ans) There are several methods for handling collisions, each of them independent of the hashing algorithm.
<u>1.Open Addressing :</u> a) Linear Probing b)Quadratic Probing c) Rehashing d) Random Probing

<u>2. Chaining</u>

[Note: Primary clustering occurs in Linear Probing and it is avoided in Quadratic Probing and Double Hashing (Rehashing)]

<u>Linear Probe :</u> In Linear Probing, when inserting a new pair whose key is k, we search the hash table bucket in the order, ht[h(k) + i] % b,0<= i <= b-1,where h is the hash function and b is the number of buckets.

It is a scheme in computer programming for resolving hash collisions of values of hash functions by sequentially searching the hash table for a free location.
<u>Quadratic Probing:</u> In Quadratic Probing, a quadratic function of I is used as the increment. The search is carried out by examining buckets h(k) , [h(k) + i^2] % b and [h(k) – i^2] % b for 1<= I <= (b-1)/2.

It is a scheme in computer programming for resolving collisions in hash tables. It is an open addressing method to handle overflows after a collision takes place in some bucket of a hash table. Quadratic probing operates by taking the original hash value and adding successive values of an arbitrary quadratic polynomial to the starting value.

<u>Random Probing:</u> In Random Probing, the search for a key , k in a hash table with b buckets is carried out by examining the buckets in the order h(k), [h(k) + s(i)] % b, 1<= I <= b-1 where s(i) is a pseudo random number. The random number generator must satisfy the property that every number from 1 to b-1 must be generated exactly once as i ranges from 1 to b-1.

<u>Double hashing</u>: It is a computer programming technique used in hash tables to resolve hash collisions, cases when two different values to be searched for produce the same hash key. It is a popular collision-resolution technique in open-addressed hash tables.

43Q) List one application of Hashing.

Ans) Hashing is used to construct the symbol table used by the language compilers.

44Q) What is the use of Friend Function?

Ans) To access the Private members of one class into another class Friend function is used.

Frequently asked Data Structures Interview Questions

Queues Data Structure – Interview Questions

How is queue different from a stack?

The difference between stacks and queues is in removing. In a stack we remove the item the most recently added; in a queue, we remove the item the least recently added.

Why is a queue called a linear data structure?

Queue is said to be linear because its elements form a sequence or a linear list. Some other examples: Array. Linked List, Stacks

Write a C program to implement a queue using two stacks.

```
#include <stdio.h>
#include <stdlib.h>
struct node
{
    int data;
    struct node *next;
};
void push(struct node** top, int data);
int pop(struct node** top);
struct queue
{
    struct node *stack1;
    struct node *stack2;
};

void enqueue(struct queue *q, int x)
{
    push(&q->stack1, x);
}

void dequeue(struct queue *q)
{
    int x;
    if (q->stack1 == NULL && q->stack2 == NULL) {
        printf("queue is empty");
        return;
    }
    }
    if (q->stack2 == NULL) {
        while (q->stack1 != NULL) {
        x = pop(&q->stack1);
        push(&q->stack2, x);
        }
    }
    x = pop(&q->stack2);
    printf("%d\n", x);
}

void push(struct node** top, int data)
{
    struct node* newnode = (struct node*) malloc(sizeof(struct node));
        if (newnode == NULL) {
            printf("Stack overflow \n");
            return;
        }
    newnode->data = data;
    newnode->next = (*top);
    (*top) = newnode;
}
int pop(struct node** top)
{
    int buf;
    struct node *t;
    if (*top == NULL) {
        printf("Stack underflow \n");
        return;
    }
    else {
        t = *top;
        buf= t->data;
        *top = t->next;
        free(t);
        return buf;
    }
}

void display(struct node *top1,struct node *top2)
{
    while (top1 != NULL) {
        printf("%d  ", top1->data);
        top1 = top1->next;
    }
    while (top2 != NULL) {
        printf("%d  ", top2->data);
        top2 = top2->next;
    }
}
int main()
{
    struct queue *q = (struct queue*)malloc(sizeof(struct queue));
    int f = 0, a;
    char ch = 'y';
    q->stack1 = NULL;
    q->stack2 = NULL;
    while (ch == 'y'||ch == 'Y') {
        printf("\n*****************Enter your choice********************\n1.enqueue\n2.
        scanf("%d", &f);
        switch(f) {
            case 1 : printf("Enter the element to be added to queue\n");
                     scanf("%d", &a);
                     enqueue(q, a);
                     break;
            case 2 : dequeue(q);
                     break;
            case 3 : display(q->stack1, q->stack2);
                     break;
            case 4 : exit(1);
                     break;
            default : printf("invalid\n");
                      break;
        }
    }
}
```

Stack – Data Structure Interview Questions

How do you implement a stack using a queue in a linked list?

```
#include<stdio.h>
#include<stdlib.h>

struct node
{
    int data;
    struct node * next;
};

struct queue
{
    struct node *rear;
    struct node *front;
};

void initial(struct queue *);
void qadd(struct queue *,int);
int qdel(struct queue *);
void dis(struct queue *);
void push(int);
void pop();

struct queue q1,q2;
int main()
{
    initial(&q1);
    initial(&q2);
    push(1);
    push(2);
    push(3);
    pop();
    printf("\nelements now are:\n");
    display(&q1);

    return 0;
}

void initial(struct queue *q)
```

```
{
    q->front=NULL;
    q->rear=NULL;
}

void qadd(struct queue *q,int n)
{
    struct node *tmp;
    tmp=(struct node *)malloc(sizeof(struct node));

    tmp->data=n;
    tmp->next=NULL;

    if(q->front==NULL)
    {
        q->rear=tmp;
        q->front=tmp;
        return;
    }

    q->rear->next=tmp;
    q->rear=tmp;
}

int qdel(struct queue *q)
{
    struct node *tmp;
    int itm;
    if(q->front==NULL)
    {
        printf("\nqueue is empty");
        return NULL;
    }

    //itm=q->front->data;
    tmp=q->front;
    itm=tmp->data;
    q->front=tmp->next;
    free(tmp);
    return itm;

}

void display(struct queue *q)
{
    struct node *tmp;
    tmp=q->front;
    while((tmp)!=NULL)
    {
        printf("\n%d",(tmp->data));
        tmp=tmp->next;
    }
    printf("\n");
}


void push(int val)
{
    struct queue tmp;
    int j;
    qadd(&q2,val);

    while(((&q1)->front)!=NULL)
    {
        j=qdel(&q1);
        qadd(&q2,j);
    }


    tmp=q1;  //swap q1 and q2
    q1=q2;
    q2=tmp;

    printf("\nelements after pushing are:\n");
    display(&q1);

}

void pop()
{
    printf("\n element deleted is %d",qdel(&q1));
}
```

Data Structures Common Interview Questions

If you are using C language to implement the heterogeneous linked list, what pointer type will you use?

The heterogeneous linked list contains different data types in its nodes and we need a link, pointer to connect them. It is not possible to use ordinary pointers for this. So we go for a void pointer. A void pointer is capable of storing a pointer to any type as it is a generic pointer type.

What factors determine the choice of data structure for a program?

As a programmer, we should consider the following factors. They are:

- The size of the data
- Time complexity
- The speed of data use
- The size of storage

What is the minimum number of queues needed to implement the priority queue?

Two. One queue is used for actual storing of data and another for storing priorities.

What are the data structures used to perform recursion?

Stack. Because of its LIFO (Last In First Out) property, it remembers its ‘caller’ so knows whom to return when the function has to return. Recursion makes use of system stack for storing the return addresses of the function calls. Every recursive function has its equivalent iterative (non-recursive) functions.For these non-recursive functions also, system stack will be used.

What are the methods available for storing sequential files?

Straight merging

Natural merging

Polyphase sort

Distribution of Initial runs.

List out few of the applications that make use of Multilinked Structures?

Sparse matrix

Index generation.

What is the type of the algorithm used in solving the 8 Queens problem?

Backtracking(stack application).

In an AVL tree, at what condition the balancing is to be done?

If the ‘Height factor’ is greater than 1 or less than -1.

What is the difference between stack and queue?

Stack follows LIFO(Last In First Out) principle whereas queue follows FIFO(First In Last Out) principle.

What are prefix and postfix notation for the expression A+B*C-D/E?

Prefix notation for the given expression: -+A*BC/DE

Postfix notation for the given expression: ABC*+DE/-

when the overlapping and collision occur at the same time, then what is the size of the bucket?

One. If there is only one entry possible in the bucket, when the collision occurs, there is no way to accommodate the colliding value which results in the overlapping of values.

In RDBMS, what is the efficient data structure used in the internal storage representation?

B+ tree because in B+ tree, all the data is stored only in leaf nodes, that makes searching easier. This represents the records that shall be stored in leaf nodes.

Whether a Linked List is linear or Non-linear data structure?

According to Accessing strategies Linked list is a linear one.

According to Storage Linked List is a Non-linear one.

What do you mean by a free pool?

A Pool is a list consisting of unused memory cells which have its own pointers.

What is Asymptotic Analysis of an algorithm?

Asymptotic analysis of an algorithm refers to defining the mathematical foundation of its run-time performance. Using this analysis, we can conclude the best case, worst case, and average case scenario of an algorithm.

What are asymptotic notations?

Asymptotic analysis can provide three levels of mathematical binding of the execution time of an algorithm:

Best case by ?(n) notation.

worst case by O(n) notation.

average case by ?(n) notation.

Name some of the approaches to develop algorithms?

There are three commonly used approaches. They are:

There are three commonly used approaches. They are:

Greedy Approach – Finding a solution by choosing the next best option.

Divide and Conquer – Divide the problem to a minimum possible sub-problem and solve them independently.

Dynamic programming – Divide the problem to a minimum possible sub-problem and solve them combinedly.

How many spanning trees can a graph have?

It basically depends on the connections of the graph. A complete undirected graph can have a maximum of n^{2}-1 spanning trees, where `n` is the number of nodes.

What is hashing?

Hashing is a technique to convert a range of key values into a range of indexes in an array. By using hash tables, we can create an associative data storage where data index can be found by providing its key value.

When does a Spanning Tree could be called as Minimum Spanning Tree?

In a Weighted Graph, a Minimum Spanning Tree is a spanning tree that has the minimum weight among all spanning trees of the same graph.

How to find middle element of linked list in one pass?

To find the middle element of the middle element of linked list in one pass, we need to maintain two pointers, one increment at each node while other increments after two nodes at a time. Through this arrangement, when the first pointer reaches the end, the second pointer will point to the middle of the linked list.

How to find if a linked list has a loop?

While finding middle element of linked list in one pass, there may be a chance that one node will be pointed by the two pointers, in this case, we can say that the linked list has a loop.

What are the different techniques for making hash functions?

Techniques for making hash functions are:

- Truncation method

- Mid square method
- Folding method
- Division method

What is Huffman's algorithm?

It is used in creating extended binary trees that have minimum weighted path lengths from the given weights. It makes use of a table that contains the frequency of occurrence for each data element.

What is a dequeue?

A dequeue is a double-ended queue. The elements can be inserted or removed into this queue from both the ends.

Can a stack be described as a pointer?

A stack can be represented as a pointer. Stack has a head pointer which points to the top of the stack. Stack operations are performed using the head pointer. Hence, the stack can be described as a pointer.

What is space complexity and time complexity?

The space complexity of a program is the amount of memory consumed by the algorithm until it completes its execution. An efficient algorithm takes space as small as possible.

The time complexity is the amount of time required by an algorithm to execute. It allows comparing the algorithm to check which one is the efficient one.

What is the difference between Binary Tree and Binary Search Tree?

Binary Tree: In a binary tree, each node can have a maximum of 2 child nodes, and there is no ordering in terms of how the nodes are organized in the binary tree. Nodes that do not have any child nodes are called leaf nodes of the binary tree.

Binary Search Tree: Binary Search Tree is essentially a binary tree, in terms of how many child nodes a node in the binary search tree can possibly have.

The important difference between a Binary Tree and a Binary Search Tree: In a binary search tree there is a relative ordering in how the nodes are organized, while there is nothing of that sort in a binary tree. In Binary search tree, all the nodes to the left of a node have values less the value of the node, and all the nodes to the right of a node have values greater than the value of the node.

What is queuing simulation?

Queuing simulation as a method is used to analyze how systems with limited resources distribute those resources to the elements waiting to be served, while waiting elements may exhibit discrete variability in demand, i.e. arrival times and require discrete processing time.

List out the applications of minimum spanning tree?

- Telephone exchanges
- Networking of computers in the lab for minimizing the length of wire.
- It provides a reasonable way for clustering points in space into natural groups.

List out the applications queues?

Two major applications of a queue are:

QueueSsimulation: It is a modeling activity used to generates statistics about the performance of the queue.

Categorization: Queues are used to categorize data into different groups without losing the order of the original data.

List out the common operations that are associated with lists?

- Insertion
- Deletion
- Retrieval
- Traversal

What is the maximum and minimum height of a binary tree having N nodes?

The maximum height of the binary tree, Hmax=N

The minimum height of a binary tree, Hmin=[log2N]+1

What are the three approaches for inserting data into general trees?

The three approaches are:

- FIFO(First In First Out)
- LIFO(Last In First Out)

- Key sequenced

What are the properties that make a binary tree as a binary search tree(BST)?

The properties of are:

- All items on the left subtrees are less than the root.
- All items in the right subtree are greater than the root or equal to the root
- Each subtree is itself a binary search tree.

Which traversal in the binary search tree will produce a sequenced list?

In order traversal in the binary search tree(BST) produces a sequential list. This traversal will give a sequential list in ascending order.

List out the applications of a heap?

The common applications of heaps are selection, priority queue, and sorting.

What are the storage structures to represent graphs?

The storage structures to represent graphs are:

Adjacency matrix

Adjacency list

What are the standard traversals in the graphs?

The two standard graph traversals: depth-first and breadth-first.

In the depth-first traversal, all of a node's descendants are processed before moving to an adjacent node.

In the breadth-first traversal, all of the adjacent vertices are processed before processing the descendants of a vertex.

What are the major data structures used in the given areas: RDBMS, Network data model, and hierarchical data model?

RDBMS –arrays

Network data model –graph

Hierarchical data model –trees

Which notations are used in the evaluation of arithmetic expressions using prefix and postfix forms?

The notations used are Polish and Reverse notation.

When is a binary search best applied?

A binary search algorithm that is best applied to search a list when the elements are already in the order or sorted.

In what areas the structures are used?

Data structures are used in every aspect where data is involved. The following areas use algorithms of data structures most efficiently. They are:

- Numerical analysis
- Artificial intelligence
- Database management structures
- Compiler designing
- Operating systems
- Networks

How does dynamic memory allocation help n managing data?

Dynamic memory allocation can combine separately allocated structured blocks to form composite structures that expand and contract as needed.

What is the primary advantage of a linked list?

The main advantage of a linked list is that it can be modified easily. The modification of a linked list works regardless of how many elements are in the list.

What is the advantage of a heap over stack?

The heap is more flexible than the stack. That’s because memory space for the heap can be dynamically allocated and de-allocated as needed.

What is an AVL tree?

An AVL tree is a search tree in which the height of the subtree differs by no more than one. It is otherwise called as a balanced binary tree. For AVL tree the balancing factor will be `-1,0,1` only.

What are the properties associated with the heaps?

The properties are:

The tree is complete or nearly complete

The key value of each node is greater than or equal to the key value in each of its descendants.

What is the efficiency of Insertion Sort, Selection Sort, and Bubble Sort?

The efficiency of the insertion sort, selection sort and bubble sort in big-O notation is $O(n^2)$.

What are the variations in the sequential search? Explain about them.

There are two variations in a sequential search. They are

Sentinel search

Probability search

Sentinel search: With this method, the condition ending the search is reduced to only one by artificially inserting the target at the end of the list.

Probability search: in this method, the list is ordered with the most probable elements at the beginning of the list and the least probable at the end.

Using which data structure format, the hash tables are storing values?

Arrays is used to store the content in hashtables.

How many stacks are required to implement a queue?

Two stacks are required to implement a queue.

For enqueue: take two stacks, say s1 and s2; perform push on s1.

For dequeue: if s2 is empty, pop all the elements from s1 and push it to s2.The last element you popped from s1 is an element to be dequeued. If s2 is not empty, then pop the top element in it.

How is an array different from a linked list?

A The size of the arrays is fixed, Linked Lists are Dynamic in size.

Inserting and deleting a new element in an array of elements is expensive, Whereas both insertion and deletion can easily be done in Linked Lists.

Random access is not allowed on Linked Listed.

Extra memory space for a pointer is required with each element of the Linked list.

Arrays have better cache locality that can make a pretty big difference in performance.

What data structure is used to implementing BFS(Breadth First Traversal) and DFS(Depth First Traversal)?

A queue is used to implement BFS.

A stack is used to implement DFS.

What is LRU caching scheme in data structures?

The LRU caching scheme is to remove the least recently used frame when the cache is full and a new page is referenced which is not there in the cache when we have given information about total pages and cache size.

What are the two data structures that are used to implement LRU caching?

A Queue is implemented using a doubly linked list. The maximum size of the queue will be equal to the total number of frames available (cache size). The most recently used pages will be near the front end and least recently pages will be near the rear end.

A Hash with page number as key and address of the corresponding queue node as value.

List out different types of loops with its efficiencies in data structures?

Following table demonstrates the efficiency trends for different data structures.

Data Structure	Efficiency
Logarithmic loop	f(n)=log n
Linear loop	f(n)=n
Linear logarithmic loop	f(n)=n(log n)
Quadratic loop	$f(n)=n^2$
Dependent quadratic loop	f(n)=n(n+1)/2

What are the two approaches that are used to write repetitive algorithms?

Iteration and recursion are the two repetitive algorithms. Recursion is a repetitive process in which an algorithm calls itself. Iteration algorithm is generally used with loops.

What is the general rule for designing a recursive algorithm?

Determine the base case.

Determine the general case

Combine the base case and general case in the algorithm

What are the two features that most affect the performance of queues?

The most important features are its arrival rate and the service time.

The rate at which the customer arrives into the queue for service is known as arrival rate.

The average time required to complete the processing of customer request is known as service time.

Data Structures

- ✦ Data Structures Tutorial
- ✦ Data Structure - Queue
- ✦ Data Structure - Stack
- ✦ Data Structure - Tree
- ✦ Data Structures - Sorting Algorithms

➪ **Data Structures Interview Questions**