

Algorithme & Data Structure

What is linear data structure?

A linear data-structure has sequentially arranged data items. It is stored and accessed in a sequential manner. Array and lists are examples of linear data structure.

What are common operations that can be performed on a data-structure?

The following operations are commonly performed on any data-structure –

- **Insertion** – adding a data item
- **Deletion** – removing a data item
- **Traversal** – accessing and/or printing all data items
- **Searching** – finding a particular data item
- **Sorting** – arranging data items in a pre-defined sequence

Briefly explain the approaches to develop algorithms.

There are three commonly used approaches to develop algorithms –

- **Greedy Approach** – finding solution by choosing next best option
- **Divide and Conquer** – dividing the problem to a minimum possible sub-problem and solving them independently
⇒ problems find their solution using divide and conquer: Merge Sort, Quick Sort, Binary Search
- **Dynamic Programming** – dividing the problem to a minimum possible sub-problem and solving them combinedly

What is stack?

In data-structure, stack is an Abstract Data Type (ADT) used to store and retrieve values in Last In First Out method

Why do we use stacks?

Stacks follow the LIFO method and addition and retrieval of a data item takes only $O(n)$ time. Stacks are used where we need to access data in the reverse order or their arrival. Stacks are used commonly in recursive function calls, expression parsing, depth first traversal of graphs etc.

What operations can be performed on stacks?

The below operations can be performed on a stack –

- **push()** – adds an item to stack
- **pop()** – removes the top stack item
- **peek()** – gives value of top item without removing it
- **isempty()** – checks if stack is empty
- **isfull()** – checks if stack is full

What is a queue in data-structure?

Queue is an abstract data structure, somewhat similar to stack. In contrast to stack, the queue is opened at both ends. One end is always used to insert data (enqueue) and the other is used to remove data (dequeue). Queue follows First-In-First-Out methodology, i.e., the data item stored first will be accessed first.

Why do we use queues?

As queues follow FIFO methods, they are used when we need to work on data-items in the exact sequence of their arrival. Every operating system maintains queues of various processes. Priority queues and breadth first traversal of graphs are some examples of queues.

Example of application: As waiting lists for a single shared resource in a printer, CPU, call center systems, or image uploads; where the first one entered is the first to be processed

What operations can be performed on Queues?

The below operations can be performed on a stack –

- **enqueue()** – adds an item to rear of the queue
- **dequeue()** – removes the item from front of the queue
- **peek()** – gives value of front item without removing it
- **isempty()** – checks if stack is empty
- **isfull()** – checks if stack is full

What is linear searching?

Linear search tries to find an item in a sequentially arranged data type. These sequentially arranged data items known as array or list, are accessible in incrementing memory location. Linear search compares expected data item with each of data items in list or array. The average case time complexity of linear search is $O(n)$ and worst case complexity is $O(n^2)$. Data in target arrays/lists need not to be sorted.

What is binary search?

A binary search works only on sorted lists or arrays. This search selects the middle which splits the entire list into two parts. First the middle is compared.

This search first compares the target value to the mid of the list. If it is not found, then it makes a decision on whether.

What is bubble sort and how bubble sort works?

Bubble sort is a comparison based algorithm in which each pair of adjacent elements is compared and elements are swapped if they are not in order. Because the time complexity is $O(n^2)$, it is not suitable for large sets of data.

Tell me something about 'insertion sort'?

Insertion sort divides the list into two sub-list, sorted and unsorted. It takes one element at a time and finds it appropriate location in the sorted sub-list and inserts it there. The output after insertion is a sorted sub-list. It iteratively works on all the elements of an unsorted sub-list and inserts them to the sorted sub-list in order.

What is selection sort?

Selection sort is in-place sorting technique. It divides the data set into two sub-lists: sorted and unsorted. Then it selects the minimum element from the unsorted sub-list and places it into the sorted list. This iterates unless all the elements from unsorted sub-list are consumed into sorted sub-list.

How insertion sort and selection sorts are different?

Both sorting techniques maintain two sub-lists, sorted and unsorted and both take one element at a time and place it into the sorted sub-list. Insertion sort works on the current element in hand and places it in the sorted array at appropriate location maintaining the properties of insertion sort. Whereas, selection sort searches the minimum from the unsorted sub-list and replaces it with the current element in hand.

What is merge sort and how it works?

Merge sort is a sorting algorithm based on divide and conquer programming approach. It keeps on dividing the list into smaller sub-list until all sub-list has only 1 element. And then it merges them in a sorted way until all sub-lists are consumed. It has run-time complexity of $O(n \log n)$ and it needs $O(n)$ auxiliary space.

What is shell sort?

Shell sort can be said to be a variant of insertion sort. Shell sort divides the list into smaller sublist based on some gap variable and then each sub-list is sorted using insertion sort. In best cases, it can perform upto $O(n \log n)$

How quicksort works?

Quick sort uses divide and conquer approach. It divides the list in smaller 'partitions' using 'pivot'. The values which are smaller than the pivot are arranged in the left partition and greater values are arranged in the right partition. Each partition is recursively sorted using quick sort.

What is a tree?

A tree is a minimally connected graph having no loops and circuits.

What is a binary tree?

A binary tree has a special condition that each node can have two children at maximum.

What is a binary search tree?

A binary search tree is a binary tree with a special provision where a node's left child must have value less than its parent's value and node's right child must have value greater than its parent value.

What is tree traversal?

Tree traversal is a process to visit all the nodes of a tree. Because, all nodes are connected via edges (links) we always start from the root (head) node. There are three ways which we use to traverse a tree –

- In-order Traversal
- Pre-order Traversal
- Post-order Traversal

What is a heap in data structure?

Heap is a special balanced binary tree data structure where the root-node key is compared with its children and arranged accordingly. A **min-heap**, a parent node has key value less than its child's and a **max-heap** parent node has value greater than its child's.

What is hashing?

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

What are linear and non linear data Structures?

- **Linear:** A data structure is said to be linear if its elements form a sequence or a linear list. Examples: Array, Linked List, Stacks and Queues

- **Non-Linear:** A data structure is said to be non-linear if traversal of nodes is nonlinear in nature. Example: Graph and Trees.

The difference between stacks and queues:

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. Both Queues and Stacks can be implemented using Arrays and Linked Lists.

What is a Linked List and What are its types?

A linked list is a linear data structure (like arrays) where each element is a separate object. Each element (that is node) of a list is comprising of two items – the data and a reference to the next node. Types of Linked List :

- **Singly Linked List:** In this type of linked list, every node stores the address or reference of the next node in list and the last node has the next address or reference as NULL. 1->2->3->4->NULL
- **Doubly Linked List:** Here, there are two references associated with each node, One of the reference points to the next node and one to the previous node. Eg. NULL<-1<->2<->3->NULL
- **Circular Linked List:** Circular linked list is a linked list where all nodes are connected to form a circle. There is no NULL at the end. A circular linked list can be a singly circular linked list or doubly circular linked list. Eg. 1->2->3->1 [The next pointer of last node is pointing to the first]

Which data structures are used for BFS and DFS of a graph?

Queue is used for BFS

Stack is used for DFS. DFS can also be implemented using recursion (recursion also uses function call stack)

Can doubly linked be implemented using a single pointer variable in every node?

Doubly linked list can be implemented using a single pointer using a **XOR linked list**. In the XOR linked list, instead of storing actual memory addresses, every node stores the XOR of addresses of previous and next nodes.

How to implement a stack using a queue?

A stack can be implemented using two queues. Let stack be implemented be 's' and queues used to implement be 'q1' and 'q2'. Stack 's' can be implemented in two ways:

- Method 1 (By making push operation costly)
- Method 2 (By making pop operation costly)

How to implement a queue using stack?

A queue can be implemented using two stacks. Let queue to be implemented be q and stacks used to implement q be stack1 and stack2. q can be implemented in two ways:

- Method 1 (By making enqueue operation costly)
- Method 2 (By making dequeue operation costly)

Which Data Structure Should be used for implementing LRU cache?

We use two data structures to implement an LRU Cache.

- **Queue** which 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 rear end and least recently pages will be near the front end.
- **A Hash** with page number as key and address of the corresponding queue node as value.

How to check if a given Binary Tree is BST or not?

If inorder traversal of a binary tree is sorted, then the binary tree is BST. The idea is to simply do inorder traversal and while traversing keep track of previous key value. If the current key value is greater, then continue, else return false.

Which data structures are applied when dealing with a recursive function?

Recursion, is a function that calls itself based on a terminating condition, makes use of the stack. Using LIFO, a call to a recursive function saves the return address so that it knows how to return to the calling function after the call terminates.

When is a binary search best applied?

A binary search is an algorithm that is best applied to search a list when the elements are already in order or sorted. The list is searched starting in the middle, such that if that middle value is not the target search key, it will check to see if it will continue the search on the lower half of the list or the higher half. The split and search will then continue in the same manner.

What is LIFO?

LIFO is a short form of Last In First Out. It refers to how data is accessed, stored and retrieved. Using this scheme, data that was stored last should be the one to be extracted first. This also means that in order to gain access to the first data, all the other data that was stored before this first data must first be retrieved and extracted.

What is FIFO?

FIFO stands for First-in, First-out, and is used to represent how data is accessed in a queue. Data has been inserted into the queue list the longest is the one that is removed first.

What are binary trees?

A binary tree is one type of data structure that has two nodes, a left node, and a right node. In programming, binary trees are an extension of the linked list structures

Are linked lists considered linear or non-linear data structures?

It depends on where you intend to apply linked lists. If you based it on storage, a linked list is considered non-linear. On the other hand, if you based it on access strategies, then a linked list is considered linear.

How do you insert a new item in a binary search tree?

Assuming that the data to be inserted is a unique value (that is, not an existing entry in the tree), check first if the tree is empty. If it's empty, just insert the new item in the root node. If it's not empty, refer to the new item's key. If it's smaller than the root's key, insert it into the root's left subtree, otherwise, insert it into the root's right subtree.

What is the minimum number of queues needed when implementing a priority queue?

The minimum number of queues needed in this case is two. One queue is intended for sorting priorities while the other queue is used for actual storage of data.

Differentiate STACK from ARRAY.

Stack follows a LIFO pattern. It means that data access follows a sequence wherein the last data to be stored when the first one to be extracted. Arrays, on the other hand, do not follow a particular order and instead can be accessed by referring to the indexed element within the array.

Basic algorithm for searching a binary search tree.

- 1. if the tree is empty, then the target is not in the tree, end search*
- 2. if the tree is not empty, the target is in the tree*
- 3. check if the target is in the root item*
- 4. if a target is not in the root item, check if a target is smaller than the root's value*
- 5. if a target is smaller than the root's value, search the left subtree*
- 6. else, search the right subtree*

What is a dequeue?

A dequeue is a double-ended queue. This is a structure wherein elements can be inserted or removed from either end.

What is an AVL tree?

An AVL tree is a type of binary search tree that is always in a state of partially balanced. The balance is measured as a difference between the heights of the subtrees from the root. This self-balancing tree was known to be the first data structure to be designed as such.

What is Fibonacci search?

Fibonacci search is a search algorithm that applies to a sorted array. It makes use of a divide-and-conquer approach that can significantly reduce the time needed in order to reach the target element

Briefly explain a recursive algorithm.

Recursive algorithm targets a problem by dividing it into smaller, manageable sub-problems. The output of one recursion after processing one sub-problem becomes the input to the next recursive process.

How do you search for a target key in a linked list?

To find the target key in a linked list, you have to apply sequential search. Each node is traversed and compared with the target key, and if it is different, then it follows the link to the next node. This traversal continues until either the target key is found or if the last node is reached.

Difference between file structure and storage structure

The main difference between file structure and storage structure is based on the memory area that is being accessed.

Storage structure: It is the representation of the data structure in the computer memory.

File structure: It is the representation of the storage structure in the auxiliary memory.

Stack overflow condition.

Overflow occurs when $top = Maxsize - 1$

If the stack is full and does not contain enough space to accept a given item to be added

Stack underflow condition.

If the stack is empty and we're gonna perform a "pop" operation .

What is the difference between a PUSH and a POP?

These are ways data is stored and retrieved.

- PUSH is used to add an item to a stack, while POP is used to remove an item.

- PUSH takes two arguments, the name of the stack to add the data to and the value of the entry to be added. POP only needs the name of the stack.

- When the stack is filled and another PUSH command is issued, you get a stack overflow error, which means that the stack can no longer accommodate the last PUSH. In POP, a stack underflow error occurs when you're trying to POP an already empty stack

The steps involved in the insertion and deletion of an element in the stack.

Push:

- Increment the variable top so that it can refer to the next memory allocation
- Copy the item to the at the array index value equal to the top
- Repeat step 1 and 2 until stack overflows

Pop:

- Store the topmost element into the an another variable
- Decrement the value of the top
- Return the topmost element

What is an array?

Arrays are defined as the collection of similar types of data items stored at contiguous memory locations. It is the simplest data structure in which each data element can be randomly accessed by using its index number.

How to reference all the elements in a one-dimension array?

It can be done by using an indexed loop such that the counter runs from 0 to the array size minus one. In this manner, you can reference all the elements in sequence by using the loop counter as the array subscript.

The advantages of Linked List over an array?

The size of a linked list can be incremented at runtime which is impossible in the case of the array.

The List is not required to be contiguously present in the main memory, if the contiguous space is not available, the nodes can be stored anywhere in the memory connected through the links.

The List is dynamically stored in the main memory and grows as per the program demand while the array is statically stored in the main memory, size of which must be declared at compile time.

The number of elements in the linked list are limited to the available memory space while the number of elements in the array is limited to the size of an array.

what pointer type should be used to implement the heterogeneous linked list

The heterogeneous linked list contains different data types, so it is not possible to use ordinary pointers for this. For this purpose, you have to use a generic pointer type like void pointer because the void pointer is capable of storing a pointer to any type.

What are the drawbacks of array implementation of Queue?

- **Memory Wastage:** The space of the array, which is used to store queue elements, can never be reused to store the elements of that queue because the elements can only be inserted at front end and the value of front might be so high that, all the space before that, can never be filled.
- **Array Size:** There might be situations in which we may need to extend the queue to insert more elements if we use an array to implement a queue. It will almost be impossible to extend the array size, therefore deciding the correct array size is always a problem in array implementation of queue.

Which data structure suits the most in the tree construction?

Queue data structure

Define the graph data structure?

A graph G can be defined as an ordered set $G(V, E)$ where $V(G)$ represents the set of vertices and $E(G)$ represents the set of edges which are used to connect these vertices. A graph can be seen as a cyclic tree, where the vertices (Nodes) maintain any complex relationship among them instead of having parent-child relations.

Mention the data structures which are used in graph implementation.

For the graph implementation, following data structures are used.

- In sequential representation, the Adjacency matrix is used.
- In Linked representation, Adjacency list is used.

What are the advantages of Binary search over linear search?

In average case, linear search takes $O(n)$ time to search a list of n elements while Binary search takes $O(\log n)$ time to search a list of n elements.

What is the difference between NULL and VOID?

Null is actually a value, whereas Void is a data type identifier.

A null variable simply indicates an empty value, whereas void is used to identify pointers as having no initial size

Can we apply the Binary search algorithm to a sorted Linked list?

No, we cannot apply the binary search algorithm to a sorted linked list because finding the index of the middle element is difficult.

When can you tell that a Memory Leak will occur?

A memory leak occurs when a program does not free a block of memory allocated dynamically.

Write the pseudocode to perform in-order traversal on a binary tree.

In-order traversal is a depth-first traversal. The method is called recursively to perform traversal on a binary tree. The code is as follows:

```
struct btnode
{
    struct btnode *left;
    struct btnode *right;
}
*root = NULL, *temp = NULL;
void inorder(struct btnode *temp)
{
    if (root == NULL)
```

```

{
    printf("Root is empty");
    return;
}
if (temp->left != NULL)
    inorder(temp->left);
if (temp->right != NULL)
    inorder(t->right);
}

```

Could you explain how variable declaration affects memory allocation?

The total amount of memory to be allocated or reserved in the case of a variable declaration depends on the data type used. For instance, declaring an integer type variable reserves 4 bytes of memory space while declaring a double variable reserves 8 bytes of the available memory.

Do you know how dynamic memory allocation helps in managing data?

Dynamic memory allocation helps in storing simple structured data types. Moreover, it can combine separately allocated structured blocks for forming composite structures that contract and expand as required.

Write the recursive C function to count the number of nodes present in a binary tree.

Answer:

```

static int counter = 0;
int countnodes(struct node *root)
{
    if(root != NULL)
    {
        countnodes(root->left);
        counter++;
        countnodes(root->right);
    }
    return counter;
}

```

Write a recursive C function to calculate the height of a binary tree.

To find the height using recursion, we find the maximum of the height of subtrees on the left and right side and then add it with the root.

```

struct node
{
    int data;
    struct node *left;
    struct node *right;
};
int height(struct node *node)
{
    if(node == NULL)
        return 0;
    else
    {
        int l_side;
        int r_side;
        l_side = height(node -> left);
        r_side = height(node -> right);
        if(l_side > r_side)
        {
            return l_side + 1;
        }
        else
            return r_side + 1;
    }
}

```

}
}

Do you know how the memory is affected by signed and unsigned numbers?

Answer: For signed numbers, the first bit is reserved for indicating whether the number is positive or negative. Hence, it has one bit less for storing the value. Unlike signed numbers, unsigned numbers have all the bits available for storing the number.

Does all declaration statements result in a fixed memory reservation?

Answer: Except for pointers, all declaration statements result in a fixed memory reservation. Instead of allocating memory for storing data, a pointer declaration results in allocating memory for storing the address of the pointer variable.

For pointers, actual memory allocation for the data happens during the runtime.

What do you understand by Infix, Prefix, and Postfix notations?

- **Infix Notation** – Operators are written between the operands. This is the standard way of writing expressions. For example, $A * (B + C) / D$

- **Postfix Notation/Reverse Polish Notation** – Operators are written after the operands, hence the name. For instance, $A B C + * D /$

- **Prefix Notation/Polish Notation** – Operators are written before the operands. $/ * A + B C D$ is the prefix notation equivalent of the aforementioned postfix notation example

Please enumerate some examples of greedy and divide and conquer algorithms.

Some examples of algorithms that follow greedy approach are:

- Dijkstra's Minimal Spanning Tree
- Graph – Map Coloring
- Graph – Vertex Cover
- Job Scheduling Problem
- Knapsack Problem
- Kruskal's Minimal Spanning Tree
- Prim's Minimal Spanning Tree

Following are some notable instances of the divide and conquer approach:

- Binary Search
- Closest Pair (or Points)
- Merge Sort
- Quick Sort
- Strassen's Matrix Multiplication

Please explain a spanning tree. What is the maximum number of spanning trees a graph can have?

A spanning tree is a subset of a graph that has all the vertices but with the minimum possible number of edges. Neither a spanning tree can be disconnected and nor does it have cycles.

The maximum number of spanning trees that a graph can have depended on how connected the graph is. A complete undirected graph with n number of nodes can have a maximum of $n-1$ number of spanning trees.

How does the Kruskal's Algorithm work?

Kruskal's algorithm treats a graph as a forest and each node in it as an individual tree.

Can you explain the Tower of Hanoi problem?

The Tower of Hanoi is a mathematical puzzle that comprises three towers (or pegs) and more than one ring. Each ring is of varying size and stacked upon one another such that the larger one is beneath the smaller one.

The goal of the Tower of Hanoi problem is to move the tower of the disk from one peg to another without breaking the properties.

Please explain an MST (Minimum Spanning Tree). Also, explain how does Prim's algorithm find a minimum spanning tree.

An MST or Minimum Spanning Tree is a spanning tree in a weighted graph that has the minimum weight of all the possible spanning trees. Each node is treated as a single tree by Prim's algorithm while adding new nodes to the spanning tree from the available graph

What are the advantages of the heap over a stack?

Generally, both heap and stack are part of memory and used in Java for different needs:

- Heap is more flexible than the stack because memory space can be dynamically allocated and de-allocated as needed
- Heap memory is used to store objects in Java, whereas stack memory is used to store local variables and function call
- Objects created in the heap are visible to all threads, whereas variables stored in stacks are only visible to the owner as private memory
- When using recursion, the size of heap memory is more whereas it quickly fill-ups stack memory

Which sorting algorithm is considered the fastest? Why?

A single sorting algorithm can't be considered best, as each algorithm is designed for a particular data structure and data set. However, the QuickSort algorithm is generally considered the fastest because it has the best performance for most inputs.

The ways to determine whether a linked list has a loop.

- Using hashing
- Using the visited nodes method (with or without modifying the basic linked list data structure)
- Floyd's cycle-finding algorithm

What is the meaning of Data Abstraction?

Data abstraction is one of the widely used tools in data structures. The goal is to break down complex entities into smaller problems and solve these by using the concepts of data structures. This provides users with the advantage of being focused on the operations and not worried about how the data is stored or represented in the memory.