1. **What do you mean by the efficiency of an algorithm? Discuss time and space complexity?**

**An algorithm's efficiency is referred to as the number of computational resources used by the algorithm. Therefore, the efficiency of an algorithm can be measured based on the usage of different resources.**

**Time Complexity: The time complexity of an algorithm quantifies the amount of time taken by an algorithm to run as a function of the length of the input. Note that the time to run is a function of the length of the input and not the actual execution time of the machine on which the algorithm is running on.**

**Space Complexity: The space complexity of an algorithm quantifies the amount of space taken by an algorithm to run as a function of the length of the input**

1. **Define ADT. Write the syntax for declaring a 2d array in the memory.**

**An ADT contains some data members and related operations that involve these data members.**

**It is called abstract because it specifies the operations without providing implementation details.**

**Int a[5][4];//it will create a 2d array having 5 rows and 4 columns**

1. **Express the following function Big Oh notation F(n)=2n3+3n+5.**

**F(n) <= cG(n)**

**2n^3 +3n + 5 <= cn^3**

**C = 3**

1. **What is data structure? Differentiate between linear and nonlinear data structure.**

**Data Structure can be defined as the group of data elements which provides an efficient way of storing and organising data in the computer so that it can be used efficiently.**

**Difference between Linear and Non-linear Data Structures:** 

| S.NO | Linear Data Structure | Non-linear Data Structure |
| --- | --- | --- |
| 1. | In a linear data structure, data elements are arranged in a linear order where each and every element is attached to its previous and next adjacent. | In a non-linear data structure, data elements are attached in hierarchically manner. |
| 2. | In linear data structure, single level is involved. | Whereas in non-linear data structure, multiple levels are involved. |
| 3. | Its implementation is easy in comparison to non-linear data structure. | While its implementation is complex in comparison to linear data structure. |
| 4. | In linear data structure, data elements can be traversed in a single run only. | While in non-linear data structure, data elements can’t be traversed in a single run only. |
| 5. | In a linear data structure, memory is not utilized in an efficient way. | While in a non-linear data structure, memory is utilized in an efficient way. |
| 6. | Its examples are: array, stack, queue, linked list, etc. | While its examples are: trees and graphs. |

1. **What do understand by stack and queue? Give real life example from each.**

**Stack is a linear data structure where insertion and deletion takes place from one end. Like :- stack of plates in hotels.**

**Queue is a linear data structure where insertion and deletion takes place from both end. Like :- queue in front of a ticket counter.**

1. **What is a multidimensional array. State the operations that can be performed on an array.**

**Multidimensional arrays are an extension of 2-D matrices and use additional subscripts for indexing. A 3-D array, for example, uses three subscripts. The first two are just like a matrix, but the third dimension represents pages or sheets of elements.**

**Operations on array**

Following are the basic operations supported by an array.

* **Traverse** − print all the array elements one by one.
* **Insertion** − Adds an element at the given index.
* **Deletion** − Deletes an element at the given index.
* **Search** − Searches an element using the given index or by the value.
* **Update** − Updates an element at the given index.

1. **Differentiate between Binary Search and Sequential (linear) Search.**

|  |  |
| --- | --- |
| **Linear Search** | **Binary Search** |
| In linear search input data need not to be in sorted. | In binary search input data need to be in sorted order. |
| It is also called sequential search. | It is also called half-interval search. |
| The time complexity of linear search **O(n)**. | The time complexity of binary search**O(log n)**. |
| Multidimensional array can be used. | Only single dimensional array is used. |
| Linear search performs equality comparisons | Binary search performs ordering comparisons |
| It is less complex. | It is more complex. |
| It is very slow process. | It is very fast process. |

**8. Write the formulas to calculate the address of an element of a 2D array in row major order. If the**

**address of A [0][0] and A [0][1] are 1000 and 1010 respectively and each element occupies 2bytes then.**

**in which order do you think the array elements are stored and why you think so?**

1. **What are the disadvantages of representing a sparse matrix in normal 2D array?**

**Representing a sparse matrix by a 2D array leads to the wastage of lots of memory. And this leads to more processing time.**

**10.State the best case and worst-case time complexity of binary search**

**algorithm.**

**Best case = O(n)**

**Worst case = O(log n)**

**1. What is the postfix form of the following prefix \*+ab-cd ?**

**2. Write the condition to test “Queue is Empty”, “Queue is Full” and “Queue contains >=1” for linear**

**queue implemen1tation in linear array.**

* 1. **if front == -1**
  2. **if rear == max-1**
  3. **if front ==0**

**3. Differentiate between stack and queue.**

**Difference between Stack and Queue Data Structures are as follows:**

| **Stacks** | **Queues** |
| --- | --- |
| **Stacks are based on the LIFO principle, i.e., the element inserted at the last, is the first element to come out of the list.** | **Queues are based on the FIFO principle, i.e., the element inserted at the first, is the first element to come out of the list.** |
| **Insertion and deletion in stacks takes place only from one end of the list called the top.** | **Insertion and deletion in queues takes place from the opposite ends of the list. The insertion takes place at the rear of the list and the deletion takes place from the front of the list.** |
| **Insert operation is called push operation.** | **Insert operation is called enqueue operation.** |
| **Delete operation is called pop operation.** | **Delete operation is called dequeue operation.** |
| **In stacks we maintain only one pointer to access the list, called the top, which always points to the last element present in the list.** | **In queues we maintain two pointers to access the list. The front pointer always points to the first element inserted in the list and is still present, and the rear pointer always points to the last inserted element.** |

**4. Define priority queue.**

A priority queue is a special type of queue in which each element is associated with a priority value. And, elements are served on the basis of their priority. That is, higher priority elements are served first.

However, if elements with the same priority occur, they are served according to their order in the queue.

**5. Define stack. Give two examples (from computer science) where you use stack. Stack is a linear data structure where insertion and deletion takes place from one end.**

1. Stacks can be used to check for the balancing of paranthesis in an expression.
2. Infix to Postfix/Prefix conversion.
3. Redo-undo features at many places such as editors, photoshop, etc.

Forward and backward feature in web browsers

**6. What is the minimum no of stacks required to implement a queue?**

**two stacks**

**In order to implement the Queue using Stack, we need to consider two stacks**

**7. What are the benefits of circular queue?**

**Advantages of Circular Queue:**

* **It provides a quick way to store FIFO data with a maximum size.**
* **Efficient utilization of the memory.**
* **Doesn't use dynamic memory.**
* **Simple implementation.**
* **All operations occur in O(1) constant time.**

1. **Write the overflow condition for circular queue.**

**If ((front == 0 && rear == max-1) || (front- rear) == 1)**

1. **Explain how stack can be represented in computers memory.**

**A stack is usually represented in computers by a block of memory cells, with the "bottom" at a fixed location, and the stack pointer holding the address of the current "top" cell in the stack.**

1. **What does it mean FRONT==REAR in a queue? Explain with example.**

**If front == rear and not equal to -1 then only one element is there**

1. **What are the types of linked lists?**

**There are four key types of linked lists:**

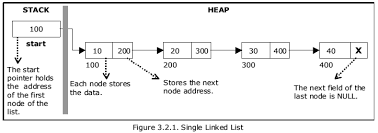
* **Singly linked lists.**
* **Doubly linked lists.**
* **Circular linked lists.**
* **Circular doubly linked lists.**

1. **How the singly linked lists can be represented?**

**Representation of Singly Linked Lists:**

**A linked list is represented by a pointer to the first node of the linked list. The first node is called the head of the linked list. If the linked list is empty, then the value of the head points to NULL**

**// make figure**

****

1. **What are the advantages of linked list?**

**Advantages Of Linked List:**

* **Dynamic data structure: A linked list is a dynamic arrangement so it can grow and shrink at runtime by allocating and**[**deallocating memory**](https://www.geeksforgeeks.org/how-to-deallocate-memory-without-using-free-in-c/)**. So there is no need to give the initial size of the linked list.**
* **No memory wastage: In the Linked list, efficient memory utilization can be achieved since the size of the linked list increase or decrease at run time so there is no memory wastage and there is no need to pre-allocate the memory.**
* **Implementation: Linear data structures like stacks and queues are often easily implemented using a linked list.**
* **Insertion and Deletion Operations: Insertion and deletion operations are quite easier in the linked list. There is no need to shift elements after the insertion or deletion of an element only the address present in the next pointer needs to be updated.**

1. **Mention the demerits of linked list**

**Demerits Of Linked List:**

* **Memory usage: More memory is required in the linked list as compared to an array. Because in a linked list, a**[**pointer**](https://www.geeksforgeeks.org/pointers-in-c-and-c-set-1-introduction-arithmetic-and-array/)**is also required to store the address of the next element and it requires extra memory for itself.**
* **Traversal: In a**[**Linked list traversal**](https://www.geeksforgeeks.org/recursive-insertion-and-traversal-linked-list/)**is more time-consuming as compared to an array. Direct access to an element is not possible in a linked list as in an array by index. For example, for accessing a node at position n, one has to traverse all the nodes before it.**
* **Reverse Traversing: In a singly linked list reverse traversing is not possible, but in the case of a**[**doubly-linked list**](https://www.geeksforgeeks.org/doubly-linked-list/)**, it can be possible as it contains a pointer to the previously connected nodes with each node. For performing this extra memory is required for the back pointer hence, there is a wastage of memory.**
* **Random Access: Random access is not possible in a linked list due to its**[**dynamic memory allocation**](https://www.geeksforgeeks.org/what-is-dynamic-memory-allocation/)**.**

1. **What are the operations performed in lists?**

**Basic Operations on Linked List**

* **Traversal : To traverse all the nodes one after another.**
* **Insertion : To add a node at the given position.**
* **Deletion : To delete a node.**
* **Searching : To search an element(s) by value.**
* **Updating : To update a node.**
* **Sorting: To arrange nodes in a linked list in a specific order.**

1. **State the Advantages of Array over Linked List**

**Advantages of Arrays:**

1. **Arrays store multiple data of similar types with the same name.**
2. **It allows random access to elements.**
3. **As the array is of fixed size and stored in contiguous memory locations there is no memory shortage or overflow.**
4. **It is helpful to store any type of data with a fixed size.**
5. **Since the elements in the array are stored at contiguous memory locations it is easy to iterate in this data structure and unit time is required to access an element if the index is known.**
6. **State the Disadvantages of Array over Linked List.**

**Disadvantages of Arrays:**

1. **The array is static in nature. Once the size of the array is declared then we can’t modify it.**
2. **Insertion and deletion operations are difficult in an array as elements are stored in contiguous memory locations and the shifting operations are costly.**
3. **The number of elements that have to be stored in an array should be known in advance.**
4. **Wastage of memory is the main problem in the array. If the array size is big the less allocation of memory leads to wastage of memory.**