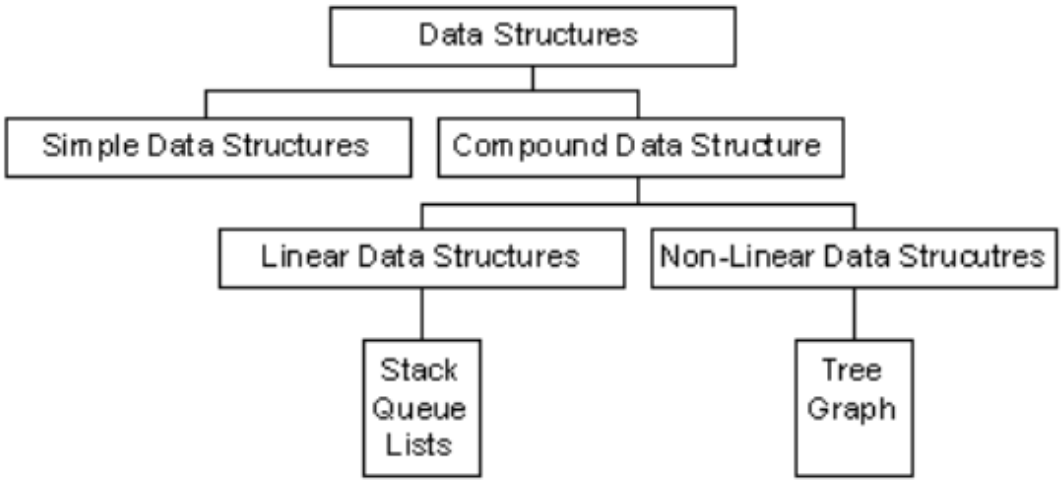
Data Structure

Data structure is a scheme for data organization so the functional definition of a data structure should be independent of its implementation. The functional definition of a data structure is known as ADT(Abstract Data Type) which is independent of implementation. The way in which the data is organized affects the performance of a program for different tasks. Computer programmers decide which data structures to use based on the nature of the data and the processes that need to be performed on that data. A data structure is a way of storing data in a computer so that it can be used efficiently and it will allow the most efficient algorithm to be used.

Data structures can be classified as:-



[Fig 1.1 Classification of Data Structures]

- Simple data structure :- Simple data structure can be constructed with the help of primitive data structure. A primitive datastructure used to represent the standard data types of any one of the computer languages.Variables,arrays,pointers,structures,unions,etc are examples of primitive data structures.

- Compound data structure :- Compound data structure can be constructed with the help of any one of the primitive data structure and it is having a specific functionality. It can be designed by user.

- Linear data structure :- Linear data structures can be constructed as a continuous arrangement of data elements in the memory.It can be constructed by using array data type. In the linear Data Structures the relationship of adjacency is maintained between the data elements.

- Non linear data structure :- Non-linear data structure can be constructed as a collection of randomly distributed set of data item joined together by using a special pointer (tag). In non-linear Data structure the relationship of adjacency is not maintained between the data items.

An algorithm is endowed with the following properties :-

1. Finiteness :- An algorithm must terminate after a finite number of steps.
2. Definiteness :- The steps of the algorithm must be precisely defined or unambiguously specified.

3. Generality :- An algorithm must be generic enough to solve all problems of a particular class.

4. Effectiveness :- The operations of the algorithm must be basic enough to be put down on pencil and paper. They should not be too complex to warrant writing another algorithm for the operation.

5. Input-Output :- The algorithm must have certain initial and precise inputs, and outputs that may be generated both at its intermediate and final steps.

Efficiency of Algorithms :- The performances of algorithms can be measured on the scales of time and space.They are described as follows :-

Time Complexity : The time complexity of an algorithm or a program is a function of the running time of the algorithm or a program. In other words, it is the amount of computer time it needs to run to completion.

Space Complexity :- The space complexity of an algorithm or program is a function of the space needed by the algorithm or program to run to completion.

Searching Techniques :-

Linear Search :- Searching is a process of finding a particular data item from a collection of data items based on specific criteria. Every day we perform web searches to locate data items containing in various pages. A search typically performed using a search key and it answers either True or False based on the item is present or not in the list. Linear search algorithm is the most simplest algorithm to do sequential search and this technique iterates over the sequence and checks one item at a time, until the desired item is found or all items have been examined.

Consider an unsorted single dimensional array of integers and we need to check whether 31 is present in the array or not, then search begins with the first element. As the first element doesn't contain the desired value, then the next element is compared to value 31 and this process continues until the desired element is found in the sixth position. Similarly, if we want to search for 8 in the same array, then the search begins in the same manner, starting with the first element until the desired element is found. In linear search, we cannot determine that a given search value is present in the sequence or not until the entire array is traversed.

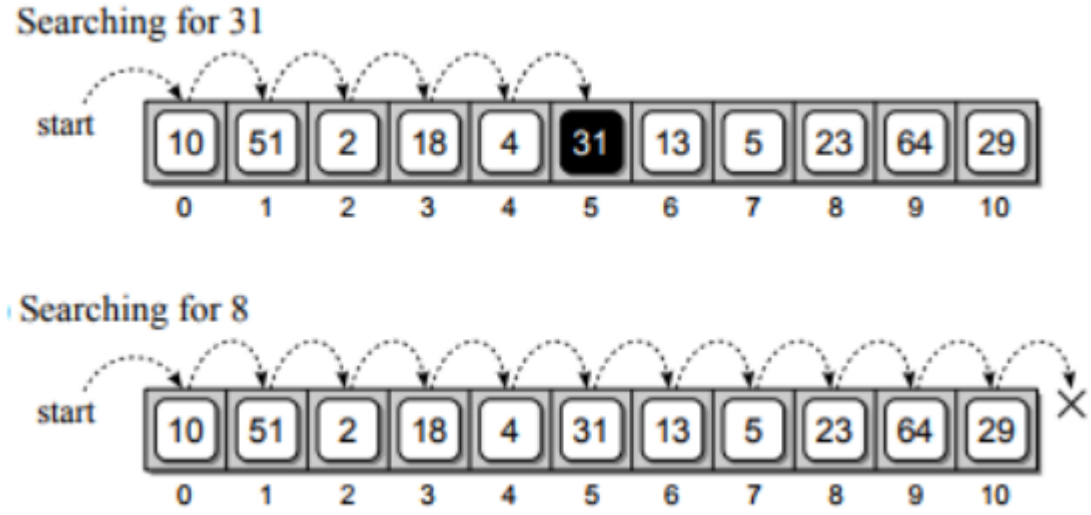


Fig:- Linear Search

Binary Search :- In Binary search algorithm, the target key is examined in a sorted sequence and this algorithm starts searching with the middle item of the sorted sequence.In binary search as the list is ordered, so we can eliminate half of the values in the list in each iteration.

Consider an example, suppose we want to search 10 in a sorted array of elements, then we first determine 14the middle element of the array. As the middle item contains 18, which is greater than the target value 10,
so can discard the second half of the list and repeat the process to first half of the array. This process is repeated until the desired target item is located in the list. If the item is found then it returns True otherwise False.

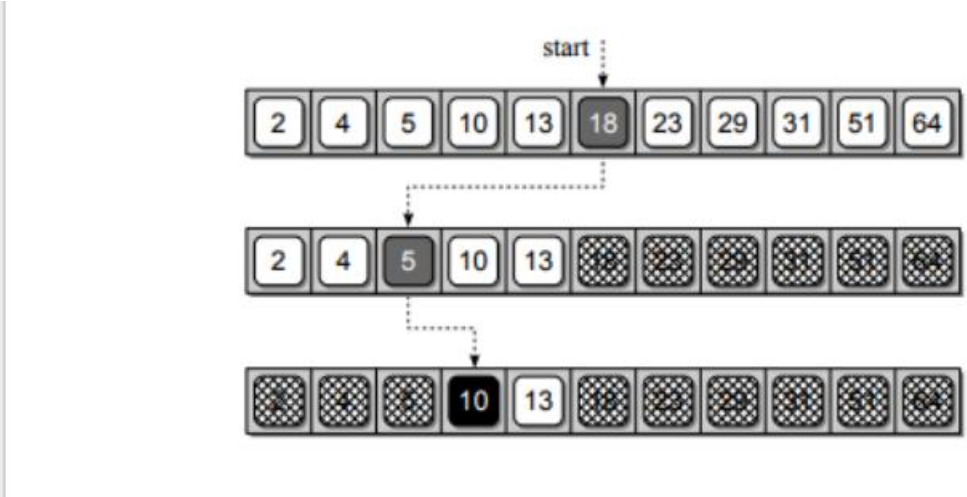


Fig:- Binary Search

Sorting Techniques :- Sorting in general refers to various methods of arranging or ordering things based on criteria's (numerical, chronological, alphabetical, hierarchical etc.). There are many approaches to sorting data and each has its own merits and demerits.

Bubble Sort :-This sorting technique is also known as exchange sort, which arranges values by iterating over the list several times and in each iteration the larger value gets bubble up to the end of the list. This algorithm uses multiple passes and in each pass the first and second data items are compared. if the first data item is bigger than the second, then the

two items are swapped. Next the items in second and third position are compared and if the first one is larger than the second, then they are swapped, otherwise no change in their order. This process continues for each successive pair of data items until all items are sorted.

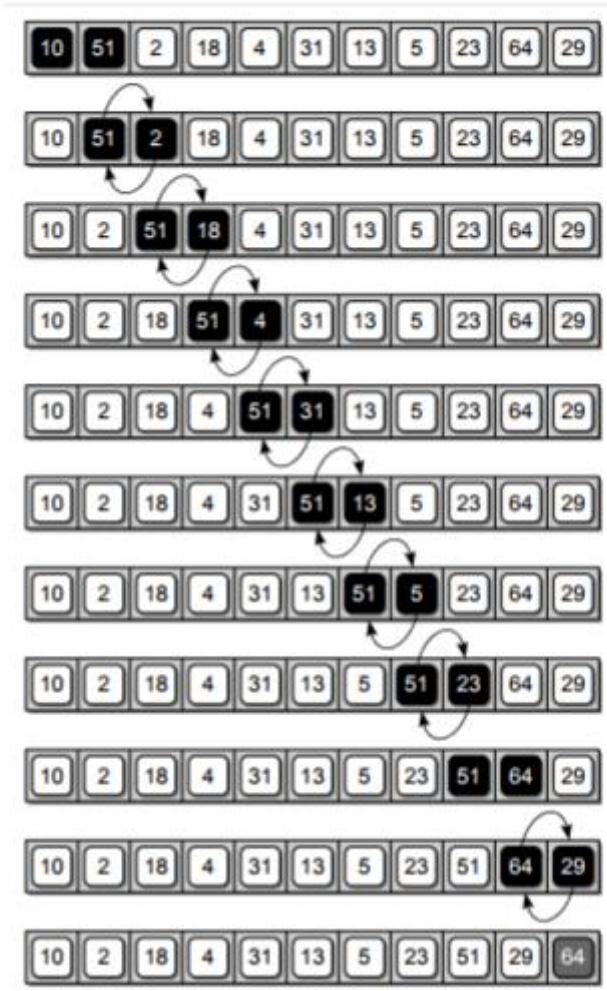


Fig:- Bubble Sort

Selection Sort :- Selection sort algorithm is one of the simplest sorting algorithm, which sorts the elements in an array by finding the minimum element in each pass from unsorted part and keeps it in the beginning. This sorting technique improves over bubble sort by making only one exchange in each pass. This sorting technique maintains two sub arrays, one sub array which is already sorted and the other one which is unsorted. In each iteration the minimum element (ascending order) is picked from unsorted array and moved to sorted sub array. An example of this sort algorithm sorting five elements:

64 25 12 22 11
11 **25** **12** 22 64
11 **12** **25** 22 64
11 12 **22** **25** 64
11 **12** **22** **25** **64**

Insertion Sort :-An algorithm consider the elements one at a time, inserting each in its suitable place among those already considered (keeping them sorted). Insertion sort is an example of an incremental algorithm. It builds the sorted sequence one number at a time. This is a suitable sorting technique in playing card games.

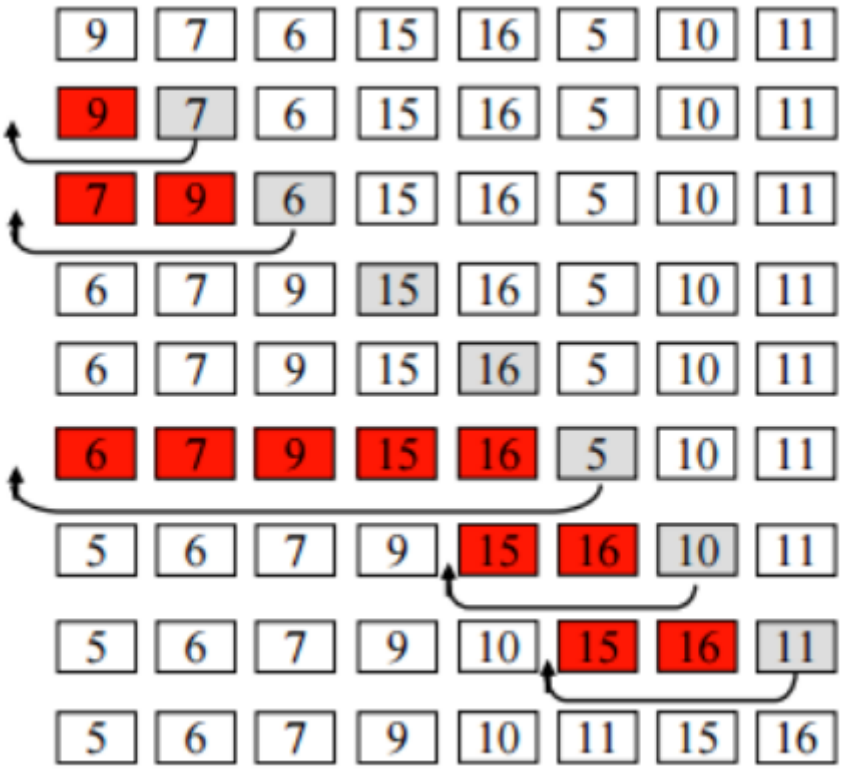


Fig:- Insertion Sort

Merge Sort :- Merge sort is based on Divide and conquer method. It takes the list to be sorted and divide it in half to create two unsorted lists. The two unsorted lists are then sorted and merged to get a sorted list. The two unsorted lists are sorted by continually calling the merge-sort algorithm; we eventually get a list of size 1 which is already sorted. The two lists of size 1 are then merged.

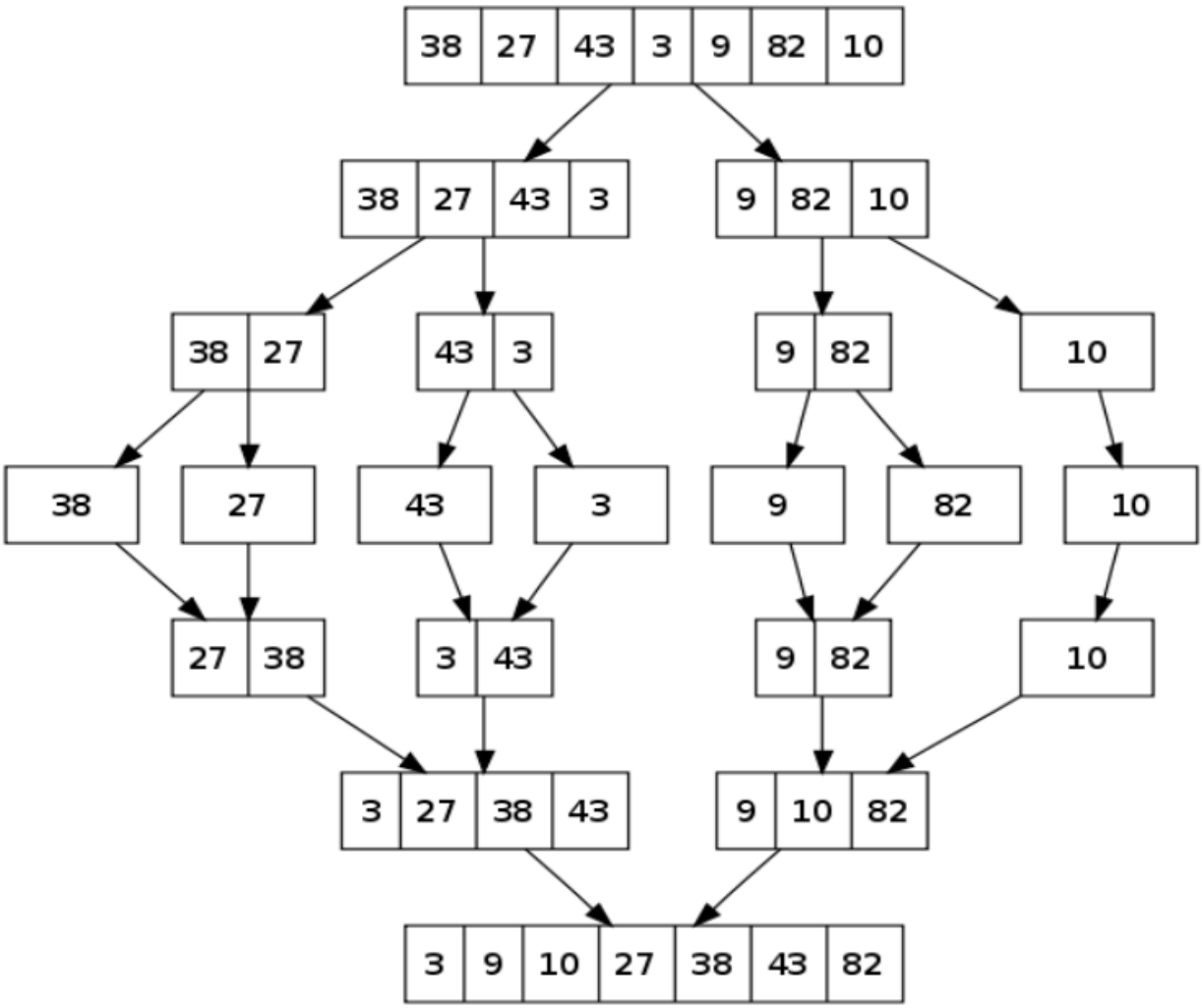


Fig:- Merge Sort

Stack :- Stack is an Abstract data structure (ADT) works on the principle Last In First Out (LIFO). The last element add to the stack is the first element to be delete. Insertion and deletion can be takes place at one end called TOP.There are two operations applied on stack,they are:-

1 . Push :- Push operation is used to add new elements in to the stack. At the time of addition first check the stack is full or not. If the stack is full it generates an error message "stack overflow".

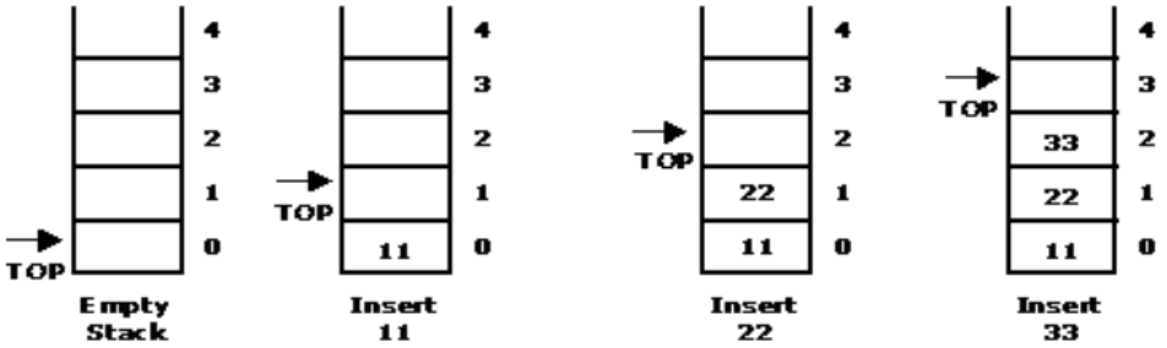


Fig:- Working of PUSH

2 . Pop :- Pop operation is used to delete elements from the stack. At the time of deletion first check the stack is empty or not. If the stack is empty it generates an error message "stack underflow".

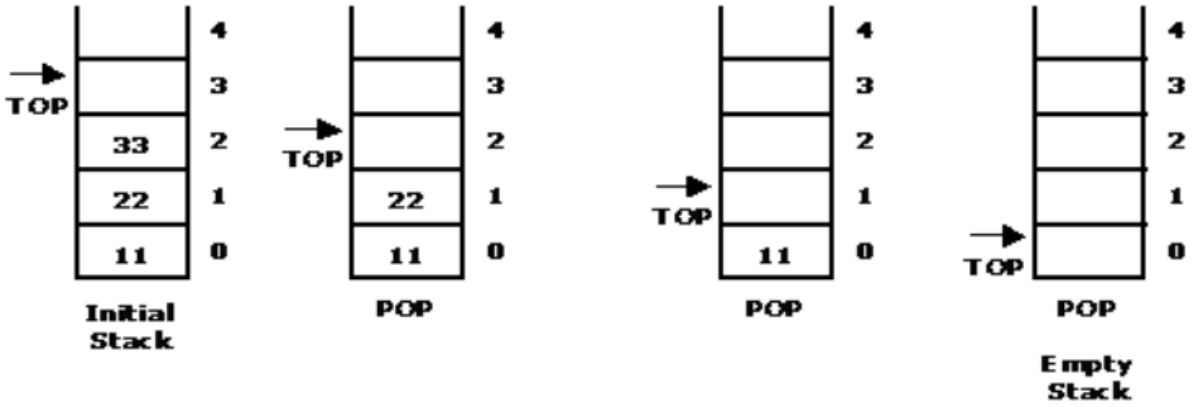


Fig:- Working of POP

Queue :- A queue is a data structure that is best described as "first in, first out". A queue is another special kind of list, where items are inserted at one end called the rear and deleted at the other end called the front. A real world example of a queue is people waiting in line at the bank. As each person enters the bank, he or she is "enqueued" at the back of the line. When a teller becomes available, they are "dequeued" at the front of the line.