Topic :- Searching

Searching means retrieving the information stored in some Data Structures like Array, LinkedList, Trees, Hash Tables etc.

Types of Searching :

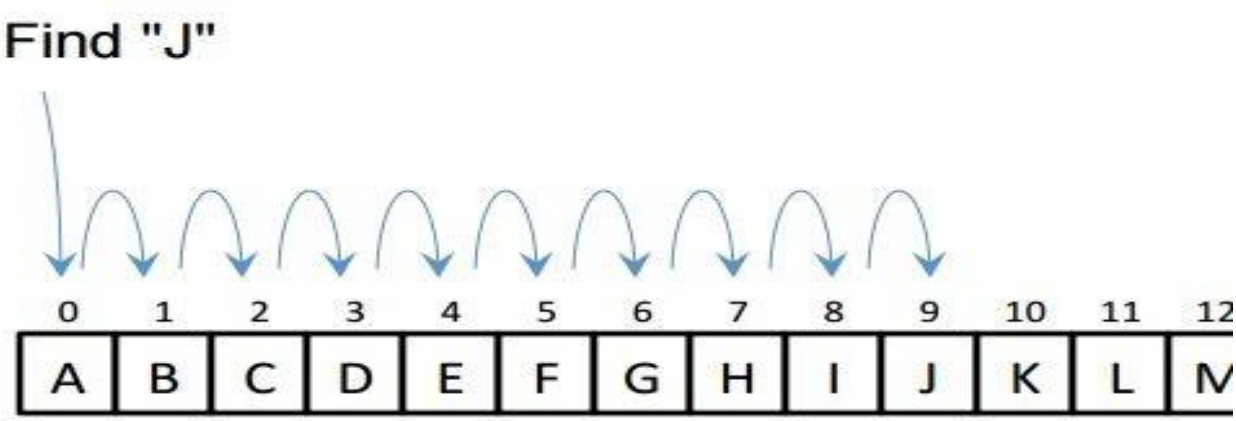
1. *Linear Search*
2. *Binary Search*
3. *Ternary Search*
4. *Jump Search*
5. *Exponential Search etc.*

1. Linear Search

Characteristics :

1. It Sequentially Searches the element.
2. It works on both sorted or unsorted Data.

➔ ***Linear Search using Array :***



❑ Iterative Approach :-

Algorithm :

1. Variable Key = User input // storing the element to be searched
2. Set variable i=0 // from where the Searching Starts from
3. If **element at i^{th} position == key element** then **break / return i** ;
   // Search terminates successfully when the element got found
4. i=i+1 // incrementing the variable i
5. If **i>=n** then **return -1** //search terminated unsuccessfully

Code :

```
1.// arr - user input array
2.// size - number of elements in array
3.// key - number to be searched
4.int linearSearch(int arr[],int size,int key){
5.     for(int i=0; i<size; i++){
6.          if(arr[i]==key)
```

```
7.              return ++i;
8.     return -1;        // return -1 when element not
  found in array
9.}
```

❑ **Recursive Approach :**

Algorithm :

1. Creating an array of n-size elements and storing values in it.
2. Storing the search element in a key variable.
3. Calling the linear_search(arr,n-1,key)
4. linear_search(array, n , target) : if element found at n^{th} position then return 'n';
5. Else return linear_search function for n-1 elements.

Code:

```
1.// arr - input array
2.// size- size of array
3.// key - number to be searched
4.int linearSearch(int arr[],int size,int key){
5.     if(size<0)          //base case
6.          return -1;
7.
8.//checking whether last element equal to key or not
```

```
9.      if(arr[size]==key)
10.         return ++size;
11.
12.    //calling recursion
13.    return linearSearch(arr,size-1,key);
14.  }
```