Searching

Searching is the process of finding some particular element in the list. If the element is present in the list, then the process is called successful and the process returns the location of that element, otherwise the search is called unsuccessful.

There are two popular search methods that are widely used in order to search some item into the list. However, choice of the algorithm depends upon the arrangement of the list.

- Linear Search
- Binary Search

Linear Search

Linear search is the simplest search algorithm and often called sequential search. In this type of searching, we simply traverse the list completely and match each element of the list with the item whose location is to be found. If the match found then location of the item is returned otherwise the algorithm return NULL.

Linear search is mostly used to search an unordered list in which the items are not sorted. The algorithm of linear search is given as follows.

Algorithm

- LINEAR_SEARCH(A, N, VAL)
- **Step 1:** [INITIALIZE] SET POS = -1
- **Step 2:** [INITIALIZE] SET I = 1
- **Step 3:** Repeat Step 4 while I<=N
- **Step 4:** IF A[I] = VAL
  SET POS = I
  PRINT POS
  Go to Step 6
  [END OF IF]
  SET I = I + 1
  [END OF LOOP]
- **Step 5:** IF POS = -1
  PRINT " VALUE IS NOT PRESENTIN THE ARRAY "
  [END OF IF]
- **Step 6:** EXIT

Complexity of algorithm

Complexity	Best Case	Average Case	Worst Case
Time	O(1)	O(n)	O(n)
Space			O(1)

C Program

```
1.  #include<stdio.h>
2.  void main ()
3.  {
4.      int a[10] = {10, 23, 40, 1, 2, 0, 14, 13, 50, 9};
5.      int item, i,flag;
6.      printf("\nEnter Item which is to be searched\n");
7.      scanf("%d",&item);
8.      for (i = 0; i< 10; i++)
9.      {
10.         if(a[i] == item)
11.         {
12.             flag = i+1;
```

```
13.             break;
14.         }
15.         else
16.         flag = 0;
17.     }
18.     if(flag != 0)
19.     {
20.         printf("\nItem found at location %d\n",flag);
21.     }
22.     else
23.     {
24.         printf("\nItem not found\n");
25.     }
26. }
```

Output:

```
Enter Item which is to be searched
20
Item not found
Enter Item which is to be searched
23
Item found at location 2
```

Java Program

```
1.  import java.util.Scanner;
2.
3.  public class Leniear_Search {
4.  public static void main(String[] args) {
5.      int[] arr = {10, 23, 15, 8, 4, 3, 25, 30, 34, 2, 19};
6.      int item,flag=0;
7.      Scanner sc = new Scanner(System.in);
8.      System.out.println("Enter Item ?");
9.      item = sc.nextInt();
10.     for(int i = 0; i<10; i++)
11.     {
12.         if(arr[i]==item)
13.         {
14.             flag = i+1;
15.             break;
16.         }
17.         else
18.             flag = 0;
19.     }
20.     if(flag != 0)
21.     {
22.         System.out.println("Item found at location" + flag);
23.     }
24.     else
25.         System.out.println("Item not found");
26.
27. }
28. }
```

Output:

```
Enter Item ?
23
Item found at location 2
Enter Item ?
22
Item not found
```

C# Program

```
1.  using System;
2.
3.  public class LinearSearch
4.  {
5.      public static void Main()
6.      {
7.          int item, flag = 0;
8.          int[]  a= {10, 23, 5, 90, 89, 34, 12, 34, 1, 78};
9.          Console.WriteLine("Enter the item value");
10.         item = Convert.ToInt32(Console.ReadLine());
11.         for(int i=0;i<10;i++)
12.         {
13.             if(item == a[i])
14.             {
15.                 flag = i+1;
16.                 break;
17.             }
18.             else
19.                 flag = 0;
20.         }
21.         if(flag != 0 )
22.         {
23.             Console.WriteLine("Item Found at Location " + flag);
24.         }
25.         else
26.             Console.WriteLine("Item Not Found");
27.
28.     }
29. }
```

Output:

```
Enter the item value
78
Item Found at Location 10

Enter the item value
22
Item not found
```

Python Program

```
1.  arr = [10,2,3,4,23,5,21,45,90,100];
2.  item = int(input("Enter the item which you want to search "));
3.  for i in range (0,len(arr)):
4.      if arr[i] == item:
5.          flag = i+1;
6.          break;
7.      else:
8.          flag = 0;
9.  if flag != 0:
10.     print("Item found at location %d" % (flag));
11. else :
12.     print("Item not found");
```

Output:

```
Enter the item which you want to search 2
Item found at location 2
Enter the item which you want to search 101
Item not found
```

Binary Search

Binary search is the search technique which works efficiently on the sorted lists. Hence, in order to search an element into some list by using binary search technique, we must ensure that the list is sorted.

Binary search follows divide and conquer approach in which, the list is divided into two halves and the item is compared with the middle element of the list. If the match is found then, the location of middle element is returned otherwise, we search into either of the halves depending upon the result produced through the match.

Binary search algorithm is given below.

BINARY_SEARCH(A, lower_bound, upper_bound, VAL)

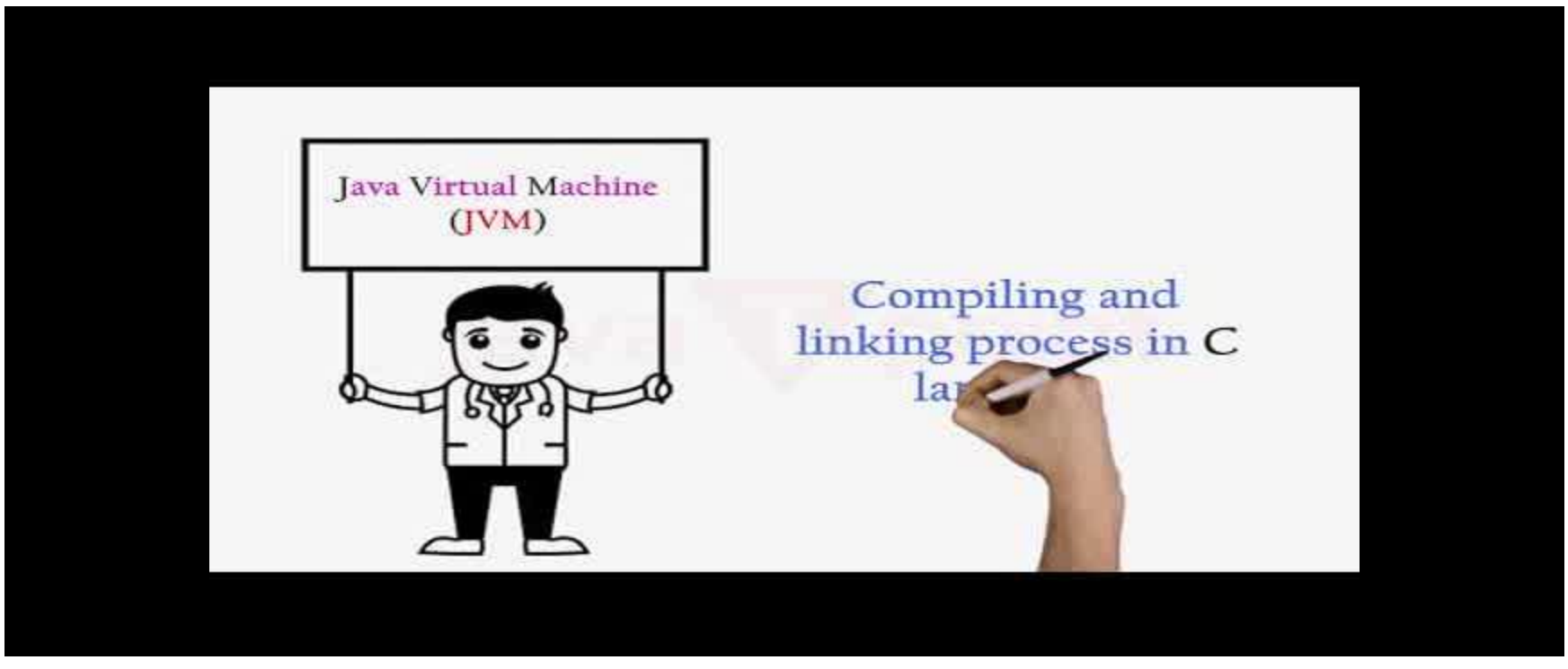
- **Step 1:** [INITIALIZE] SET BEG = lower_bound
  END = upper_bound, POS = - 1
- **Step 2:** Repeat Steps 3 and 4 while BEG <=END
- **Step 3:** SET MID = (BEG + END)/2
- **Step 4:** IF A[MID] = VAL
  SET POS = MID
  PRINT POS
  Go to Step 6
  ELSE IF A[MID] > VAL
  SET END = MID - 1
  ELSE
  SET BEG = MID + 1
  [END OF IF]
  [END OF LOOP]
- **Step 5:** IF POS = -1
  PRINT "VALUE IS NOT PRESENT IN THE ARRAY"
  [END OF IF]
- **Step 6:** EXIT

Complexity

SN	Performance	Complexity
1	Worst case	O(log n)
2	Best case	O(1)
3	Average Case	O(log n)
4	Worst case space complexity	O(1)

Example

Let us consider an array arr = {1, 5, 7, 8, 13, 19, 20, 23, 29}. Find the location of the item 23 in the array.



In 1st step :

1. BEG = 0
2. END = 8ron
3. MID = 4
4. a[mid] = a[4] = 13 < 23, therefore

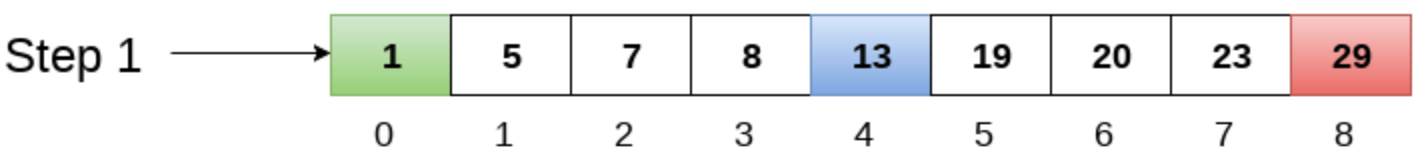
in Second step:

1. Beg = mid +1 = 5
2. End = 8
3. mid = 13/2 = 6
4. a[mid] = a[6] = 20 < 23, therefore;

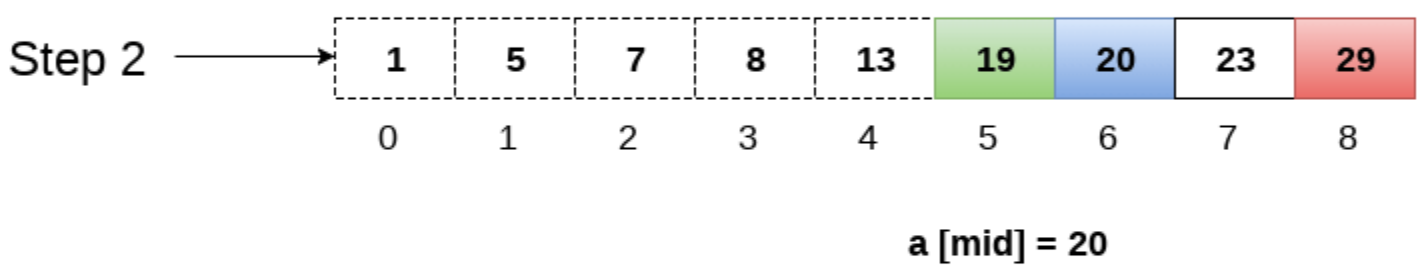
in third step:

1. beg = mid + 1 = 7
2. End = 8
3. mid = 15/2 = 7
4. a[mid] = a[7]
5. a[7] = 23 = item;
6. therefore, set location = mid;
7. The location of the item will be 7.

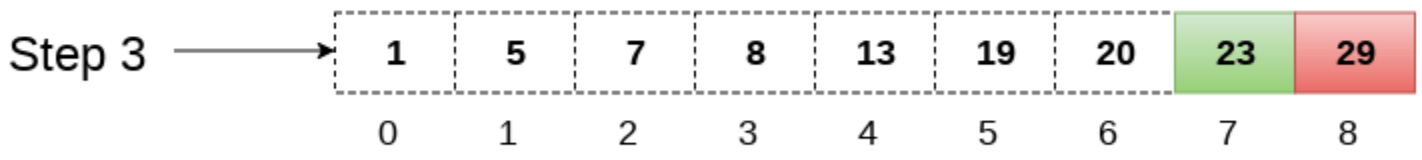
Item to be searched = 23



a [mid] = 13
13 < 23
beg = mid + 1 = 5
end = 8
mid = (beg + end)/2 = 13 / 2 = 6



a [mid] = 20
20 < 23
beg = mid + 1 = 7
end = 8
mid = (beg + end)/2 = 15 / 2 = 7



a [mid] = 23
23 = 23
loc = mid

Return location 7

Binary Search Program using Recursion

C program

```
1. #include<stdio.h>
2. int binarySearch(int[], int, int, int);
3. void main ()
4. {
5.     int arr[10] = {16, 19, 20, 23, 45, 56, 78, 90, 96, 100};
6.     int item, location=-1;
7.     printf("Enter the item which you want to search ");
8.     scanf("%d",&item);
9.     location = binarySearch(arr, 0, 9, item);
10.    if(location != -1)
11.    {
12.        printf("Item found at location %d",location);
13.    }
14.    else
15.    {
16.        printf("Item not found");
17.    }
18. }
19. int binarySearch(int a[], int beg, int end, int item)
20. {
```

```
    int mid;
    if(end >= beg)
    {
        mid = (beg + end)/2;
        if(a[mid] == item)
        {
            return mid+1;
        }
        else if(a[mid] < item)
        {
            return binarySearch(a,mid+1,end,item);
        }
        else
        {
            return binarySearch(a,beg,mid-1,item);
        }

    }
    return -1;
}
```

Output:

```
Enter the item which you want to search
19
Item found at location 2
```

Java

```
import java.util.*;
public class BinarySearch {
public static void main(String[] args) {
    int[] arr = {16, 19, 20, 23, 45, 56, 78, 90, 96, 100};
    int item, location = -1;
    System.out.println("Enter the item which you want to search");
    Scanner sc = new Scanner(System.in);
    item = sc.nextInt();
    location = binarySearch(arr,0,9,item);
    if(location != -1)
    System.out.println("the location of the item is "+location);
    else
        System.out.println("Item not found");
    }
public static int binarySearch(int[] a, int beg, int end, int item)
{
    int mid;
    if(end >= beg)
    {
        mid = (beg + end)/2;
        if(a[mid] == item)
        {
            return mid+1;
        }
        else if(a[mid] < item)
        {
            return binarySearch(a,mid+1,end,item);
        }
        else
        {
```

```
            return binarySearch(a,beg,mid-1,item);
        }

    }
    return -1;
}
}
```

Output:

```
Enter the item which you want to search
45
the location of the item is 5
```

C#

```
using System;

public class LinearSearch
{
    public static void Main()
    {
    int[] arr = {16, 19, 20, 23, 45, 56, 78, 90, 96, 100};
    int location=-1;
    Console.WriteLine("Enter the item which you want to search ");
    int item = Convert.ToInt32(Console.ReadLine());
    location = binarySearch(arr, 0, 9, item);
    if(location != -1)
    {
        Console.WriteLine("Item found at location "+ location);
    }
    else
    {
        Console.WriteLine("Item not found");
    }
}
public static int binarySearch(int[] a, int beg, int end, int item)
{
    int mid;
    if(end >= beg)
    {
        mid = (beg + end)/2;
        if(a[mid] == item)
        {
            return mid+1;
        }
        else if(a[mid] < item)
        {
            return binarySearch(a,mid+1,end,item);
        }
        else
        {
            return binarySearch(a,beg,mid-1,item);
        }

    }
    return -1;

    }
```

```
44. }
```

Output:

```
Enter the item which you want to search
20
Item found at location 3
```

Python

```
1.  def binarySearch(arr,beg,end,item):
2.      if end >= beg:
3.          mid = int((beg+end)/2)
4.          if arr[mid] == item :
5.              return mid+1
6.          elif arr[mid] < item :
7.              return binarySearch(arr,mid+1,end,item)
8.          else:
9.              return binarySearch(arr,beg,mid-1,item)
10.     return -1
11.
12.
13. arr=[16, 19, 20, 23, 45, 56, 78, 90, 96, 100];
14. item = int(input("Enter the item which you want to search ?"))
15. location = -1;
16. location = binarySearch(arr,0,9,item);
17. if location != -1:
18.     print("Item found at location %d" %(location))
19. else:
20.     print("Item not found")
```

Output:

```
Enter the item which you want to search ?
96
Item found at location 9

Enter the item which you want to search ?
101
Item not found
```

Binary Search function using Iteration

```
1.  int binarySearch(int a[], int beg, int end, int item)
2.  {
3.      int mid;
4.      while(end >= beg)
5.      {
6.          mid = (beg + end)/2;
7.          if(a[mid] == item)
8.          {
9.              return mid+1;
10.         }
11.         else if(a[mid] < item)
12.         {
13.             beg = mid + 1;
14.         }
15.         else
16.         {
17.             end = mid - 1;
18.         }
19.
20.     }
21.     return -1;
22. }
```