SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641 107
Accredited by NAAC-UGC with ‘A’ Grade
Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

Department of Artificial Intelligence and Data Science

Course Name: 23ITB201 Data structures and Algorithms

II Year / III semester

Unit III – Searching, Sorting and Hashing

Topic: Linear and Binary search

Linear 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.

Linear search

C Program for Linear search

```
#include<stdio.h>
void main ()
{
    int a[10] = {10, 23, 40, 1, 2, 0, 14, 13, 50, 9};
    int item, i,flag;
    printf("\nEnter Item which is to be searched\n");
    scanf("%d",&item);
    for (i = 0; i< 10; i++)
    {
        if(a[i] == item)
        {
            flag = i+1;
            break;
        }
        else
        flag = 0;
    }
    if(flag != 0)
        {
    printf("\nItem found at location %d\n",flag);
        }
        else
        {
            printf("\nItem not found\n");
        }
}
```

Binary Search

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

BINARY_SEARCH(A, lower_bound, upper_bound, VAL)

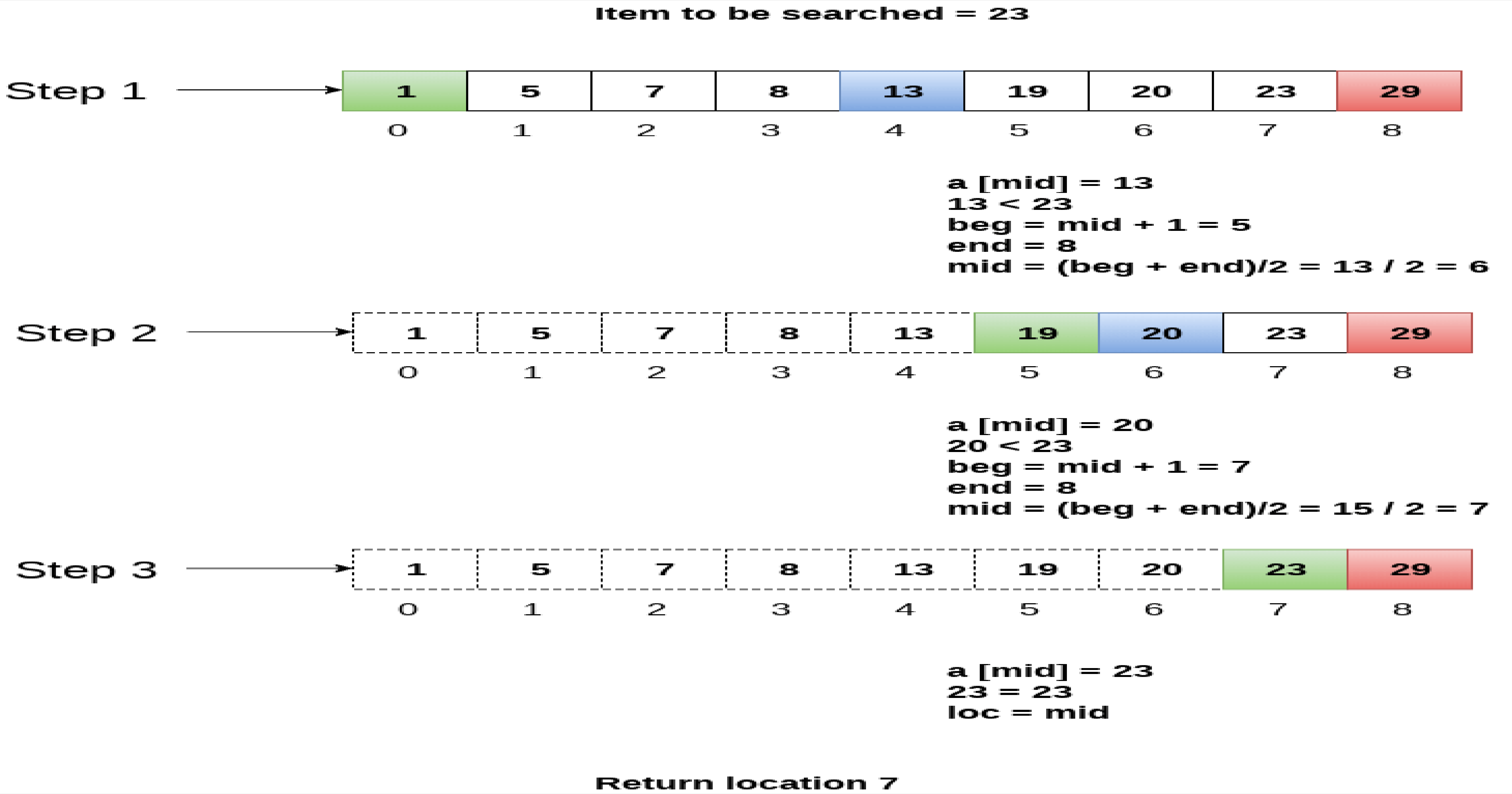
```
Step 1: [INITIALIZE] SET BEG = lower_bound
    END = upper_bound, POS = - 1
Step 2: Repeat Steps 3 and 4 while BEG <=END
Step 3: SET MID = (BEG + END)/2
Step 4: IF A[MID] = VAL
    SET POS = MID
    PRINT POS
    Go to Step 6
    ELSE IF A[MID] > VAL
    SET END = MID - 1
    ELSE
    SET BEG = MID + 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
```

Binary Search

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.

Binary Search



Item to be searched = 23

Step 1 →

1	5	7	8	13	19	20	23	29
0	1	2	3	4	5	6	7	8

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

Step 2 →

1	5	7	8	13	19	20	23	29
0	1	2	3	4	5	6	7	8

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

Step 3 →

1	5	7	8	13	19	20	23	29
0	1	2	3	4	5	6	7	8

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

Return location 7