3.2.2 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.

Let us consider an array arr = {1, 5, 7, 8, 13, 19, 20, 23, 29}. Find the location of the item23 in the array.

In 1ˢᵗ step :

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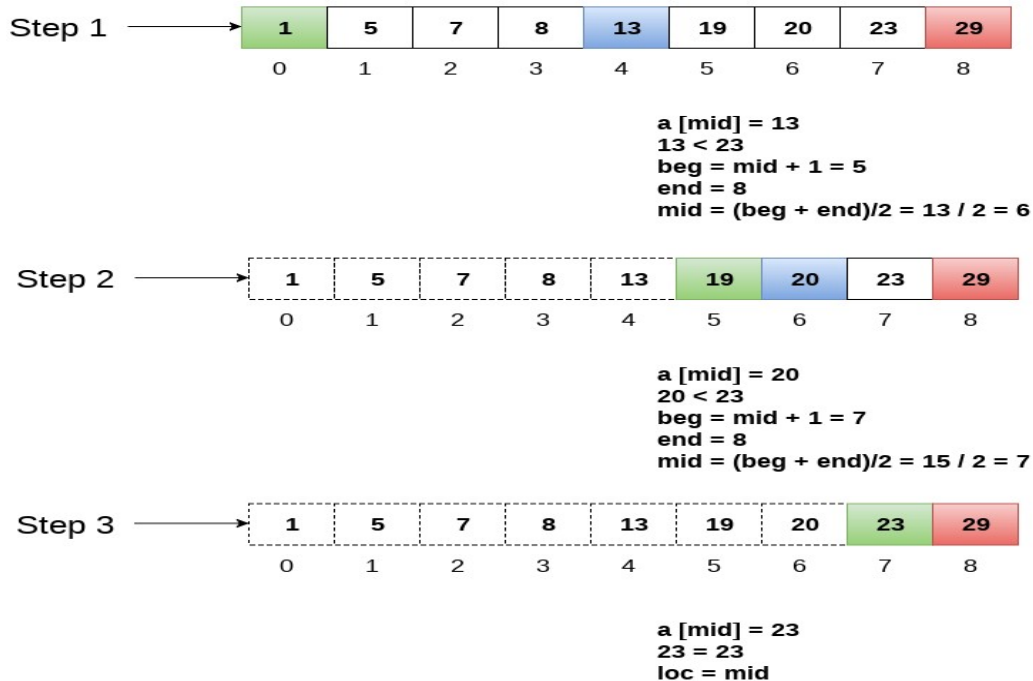
IN 2ⁿᵈ Step

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

Binary search algorithm is given below.

Algorithm for binary search

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

The algorithm for binary search.

In Step 1, we initialize the value of variables, BEG, END, and POS.
 In Step 2, a while loop is executed until BEG is less than or equal to END.
In Step 3, the value of MID is calculated.
 In Step 4, we check if the array value at MID is equal to VAL (item to be searched in the array). If a match is found, then the value of POS is printed and the algorithm exits. However, if a match is not found, and if the value of A[MID] is greater than VAL, the value of END is modified, otherwise if A[MID] is greater than VAL, then the value of BEG is altered.
In Step 5, if the value of POS = –1, then VAL is not present in the array and an appropriate message is printed on the screen before the algorithm exits.

Advantages-

 The advantages of binary search algorithm are-

- It eliminates half of the list from further searching by using the result of each comparison.
- It indicates whether the element being searched is before or after the current position in the list.
- This information is used to narrow the search.
- For large lists of data, it works significantly better than linear search.

Disadvantages-

 The disadvantages of binary search algorithm are-

- It employs recursive approach which requires more stack space.
- Programming binary search algorithm is error prone and difficult.
- The interaction of binary search with memory hierarchy i.e. caching is poor.
  (because of its random access nature)

Complexity of Binary Search Algorithm

As we dispose off one part of the search case during every step of binary search, and perform the search operation on the other half, this results in a worst case time complexity of O(log2N).

Comparison of Searching methods in Data Structures

Sequential Search	Binary Search
Time complexity is O(n)	Time complexity is O(log n)
Finds the key present at first position in constant time	Finds the key present at centre position in constant time
Sequence of elements in the container does not affect.	The elements must be sorted in the container
Arrays and linked lists can be used to implement this	It cannot be implemented directly into the linked list. We need to change the basic rules of the list to implement this
Algorithm is iterative in nature	Algorithm technique is Divide and Conquer.
Algorithm is easy to implement, and requires less amount of code.	Algorithm is slightly complex. It takes more amount of code to implement.
N number of comparisons are required for worst case.	Log n number of comparisons are sufficient in worst case.