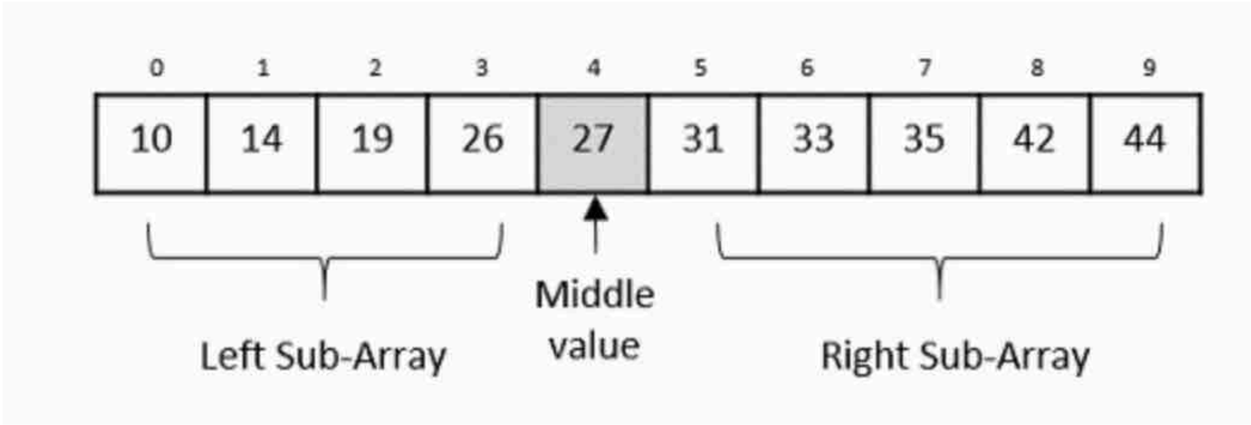
Practical 1.1

Aim: Implementation of Binary search algorithm using Divide & Conquer method.

Theory:

Binary search is an efficient algorithm for searching that has a time complexity of (log n). The working of this search is completely based on the divide and conquer technique as it splits the array in half before looking for the desired item. The data used for this algorithm must be sorted beforehand.

The process of a binary search involves finding the value of a certain key by making comparisons with the item located in the middle of the collection. In case of a positive match, the index of the item is returned. Conversely, if the item in the middle is found to be higher than the key value then the search will move to the right sub-array. If the middle value is lower than the key value then the search will go to the left sub-array. The same process will be repeated until the subarray size is reduced to zero.



Binary Search Algorithm

Binary Search algorithm is an interval searching method that performs the searching in intervals only. The input taken by the binary search algorithm must always be in a sorted array since it divides the array into subarrays based on the greater or lower values. The algorithm follows the procedure below −

Step 1 − Select the middle item in the array and compare it with the key value to be searched. If it is matched, return the position of the median.

Step 2 − If it does not match the key value, check if the key value is either greater than or less than the median value.

Step 3 − If the key is greater, perform the search in the right sub-array; but if the key is lower than the median value, perform the search in the left sub-array.

Step 4 − Repeat Steps 1, 2 and 3 iteratively, until the size of sub-array becomes 1.

Step 5 − If the key value does not exist in the array, then the algorithm returns an unsuccessful search.

Pseudocode

The pseudocode of binary search algorithms should look like this −

```
BEGIN

  READ n

  READ array elements (sorted)

  READ key

  low ← 0

  high ← n - 1

  WHILE low ≤ high DO

      mid ← (low + high) / 2

      IF array[mid] = key THEN

          PRINT "Element found at index", mid

          STOP

      ELSE IF key < array[mid] THEN
      high ← mid - 1
    ELSE
      low ← mid + 1
    END IF
  END WHILE
  PRINT "Element not found"
END
```

Explanation:

This pseudocode implements the Binary Search algorithm using the Divide and Conquer technique. First, the number of elements and the sorted array are read from the user, followed by the key element to be searched. Two variables, low and high, are initialized to represent the first and last indices of the array. The algorithm repeatedly calculates the middle index and compares the middle element with the key. If both are equal, the search is successful and the index is displayed. If the key is smaller, the search continues in the left half; otherwise, it proceeds in the right half. The process repeats until the element is found or the search range becomes empty.

Conclusion:

Binary Search efficiently finds an element in a sorted array using divide and conquer with logarithmic time complexity.