KIET Group of Institutions, Ghaziabad

Department of Computer Applications

(An ISO – 9001: 2015 Certified & 'A' Grade accredited Institution by NAAC)

Design and Analysis of Algorithm

RCA 352: Session 2020-21

DAA Lab

Experiment-No.

Objective: Implement the **binary search** algorithm to sort the given list of N numbers and plot graph

Scheduled Date:	Compiled Date:	Submitted Date:
09-10-20	09-10-20	09-10-20

Binary Search Algorithm

Algorithm

IterativeBinarySearch(Input: Array A, Start,End, K)

Start: Starting index of array A

A: Array of Size N

K:Element that you want to search

End:Last index of array A

1. mid=(Start+End)/2

2. while(Start<=End)

3. if(A[middle]==K)

4. return middle

5. elseif(A[middle] < k)

6. start=middle+1;

7. else

8. end=middle-1;

9. end of while loop

9. return -1

RecursiveBinarySearch(Input: Array A, Start,End, K)

1. if(Start<=End)

2. middle=(Start+End)/2

3. if(a[middle] == element)

4. return middle

5. else if(a[middle] < element)

6. return RecursiveBinarySearch(a,midddle+1,end,element)

7. else

8. return RecursiveBinarySearch(a,start,middle-1,element)

9. endif

10.return -1

Program of IterativeBinarySearch

```
#include<stdio.h>

#include<conio.h>

int count=0;

int iterativebinarysearch(int [],int,int,int);

int main()

{

 int arr[50],n,i,element,location;

 printf("Enter the size of array:\n");

 scanf("%d",&n);

 printf("Enter the array elements:\n");

  for(i=0;i<n;i++)

  {

   scanf("%d",&arr[i]);
```

```
 }
 printf("Enter the element to be  searched :\n");
 scanf("%d",&element);
 loc=iterative_binarysearch(arr,0,n-1,element);
  if(loc>=0)
  {
   printf("\nElement is available at position %d",location+1);
  }
  else
  {
   printf("\nElement is not available in array");
  }
  printf("for n=%d no. of counts are=%d",n,count);
getch();
return 0;
}
int iterativebinarysearch(int arr[], int start, int end, int element)
{
 int middle;
 count++;
 while(start<=end)
 {
  count++;
  middle=(start+end)/2;
  count++;
```

```
 if(arr[middle]==element)
 {
  count++;
  return middle;

 }
 else if(arr[middle]<element)
 {
  count++;
  start=middle+1;
  count++;
 }
 else
  count++;
  end=middle-1;
  count++;
}
count++;
return -1;
}
```

Binary Search Graph

Input size	Best case	Average case	Worst case
5	14	29	29

KIET Group of Institutions, Ghaziabad
Department of Computer Applications
(An ISO – 9001: 2015 Certified & 'A' Grade accredited Institution by NAAC)

Design and Analysis of Algorithm

RCA 352: Session 2020-21

DAA Lab

11	14	31	41
15	14	41	41
21	14	46	46
25	14	46	46