ASSIGNMENT NO:11 **DATE:10/04/2015**

PROGRAM TITLE:Search an Array using Linear Search, Binary Search and Interpolation Search Techniques.

THEORY: Linear Search is also known as Sequential Search. It is the most basic search as it searches through the whole array until the element is found.

Binary Search searches for the element at the middle position of the array which is calculated as mid =(lb+ub)/2; lb and ub being the lower and upper bound of the array that it is currently working on. It searches either the lower half or the upper half of the array the next time depending on whether the item to be searched is higher than the element at the middle position.

Interpolation Search works in the same basic principle as Binary Search. The only difference is in the calculation of the middle position which is calculated as $mid=lb+((ub-lb)*((item-a_{lb}))/(a_{ub}-a_{lb}))$; where item is the value to be searched for. This search is even more efficient than binary search if the elements are uniformly distributed throughout the array.

Binary and Interpolation Searches have a prerequisite of the array elements being in the sorted order.

PROGRAM ALGORITHM:

```
Algo_linsearch(a,n,key)
{
    i=1;
    $a_{n+1}$=key;
    while($a_i \neq$key)
    {
        i=i+1;
    }
    if(i$\leq$n)
        return i; //returns the position where the element is found
    else
        return -1;
}

Algo_binsearch_rec(a,lb,ub,item)
{
    if(lb>ub)
        return -1;
    else
    {
        mid=$\lfloor$(lb+ub)/2$\rfloor$;
        if(item=a[mid])
            return mid;
        else if(item<a[mid])
            return binsearchrec(a,lb,mid-1,item);
        else
            return binsearchrec(a,mid+1,ub,item);
    }
}
Algo_interpolation_search(a,lb,ub,item)
```

```
{
      do
      {
            if(mid=$\lfloor lb+((ub-lb)*((item-a_{lb}))/(a_{ub}-a_{lb}))\rfloor$)
                  break;
            mid=$\lfloor lb+((ub-lb)*((item-a_{lb}))/(a_{ub}-a_{lb}))\rfloor$;
            if(item<$a_{mid}$)
                  ub=mid-1;
            else
                  lb=mid+1;
      }
      while(($a_{mid}\neq$item)&&(lb<=ub));
      if(item=$a_{mid}$)
            return mid;
      else
            return -1;
}
```

PROGRAM CODE:

```
//C Program for Menu Driven Searching
#include <stdio.h>
int linsearch(int * a,int n,int key);//Function that searches using
linear search method
int binsearchrec(int *a,int lb,int ub,int item);// Recursive Function
that searches for given item using binary search
int interpolationsearch(int *a,int lb,int ub,int item);//Function that
searches for given item using interpolation search
int main()
{
      int n,i,key,ch;
      printf("\n\tEnter the number of elements::");
      scanf("%d",&n);
      int a[n+1];
      printf("\n\tEnter the elements::");
      for(i=0;i<n;)
      {
            scanf("%d",&a[i++]);
      }
      do
      {
            printf("\n\tEnter item to be searched::");
            scanf("%d",&key);
            printf("\n\tMenu::\n\t1.Search using Linear
Search.\n\t2.Search using Binary Search.\n\t3.Search using Interpolation
Search.\n\t4.Exit.\n\tThe Array must be in ascending order for using
Binary and Interpolation Searches.\n\tEnter Choice::");
            scanf("%d",&ch);
            switch(ch)
            {
                  case 1:i=linsearch(a,n,key);
                        break;
                  case 2: i=binsearchrec(a,0,n-1,key);
                        break;
                  case 3: i=interpolationsearch(a,0,n-1,key);
```

```
                    break;
                case 4: printf("\n\tProgram Terminated.\n");
                    return 1;
                    break;
                default:printf("\n\tInvalid Choice.  Enter Again.\n");
            }
            if(i==-1)
                printf("\n\tThe item is not found\n");
            else
                printf("\n\tThe item is found at index %d\n",i);
        }
        while(ch!=4);
        return 0;
}
int linsearch(int * a,int n,int key)
{
        int i=0;
        a[n+1]=key;
        while(a[i]!=key)
            i=i+1;
        if(i<n)
            return i;
        else
            return -1;
}
int binsearchrec(int *a,int lb,int ub,int item)
{
        if(lb>ub)
                return -1;
          else
          {
            int mid=(lb+ub)/2;
            if(item==a[mid])
                return mid;
            else if(item<a[mid])
                return binsearchrec(a,lb,mid-1,item);
            else
                return binsearchrec(a,mid+1,ub,item);
          }
}
int interpolationsearch(int *a,int lb,int ub,int item)
{
        int mid=0;
        do
        {
            if(mid==lb+((ub-lb)*((item-a[lb]))/(a[ub]-a[lb])))
                break;
            mid=lb+((ub-lb)*((item-a[lb]))/(a[ub]-a[lb]));
            if(item<a[mid])
                ub=mid-1;
            else
                lb=mid+1;
        }
        while((a[mid]!=item)&&(lb<=ub));
        if(item==a[mid])
```

```
            return mid;
        else
            return -1;
}
```

OUTPUT:

```
        Enter the number of elements::6

        Enter the elements::1 3 5 7 9 10

        Enter item to be searched::3

        Menu::
        1.Search using Linear Search.
        2.Search using Binary Search.
        3.Search using Interpolation Search.
        4.Exit.
        The Array must be in ascending order for using Binary and
Interpolation Searches.
        Enter Choice::1

        The item is found at index 1

        Enter item to be searched::9

        Menu::
        1.Search using Linear Search.
        2.Search using Binary Search.
        3.Search using Interpolation Search.
        4.Exit.
        The Array must be in ascending order for using Binary and
Interpolation Searches.
        Enter Choice::2

        The item is found at index 4

        Enter item to be searched::8

        Menu::
        1.Search using Linear Search.
        2.Search using Binary Search.
        3.Search using Interpolation Search.
        4.Exit.
        The Array must be in ascending order for using Binary and
Interpolation Searches.
        Enter Choice::3

        The item is not found

        Enter item to be searched::4

        Menu::
        1.Search using Linear Search.
        2.Search using Binary Search.
```

```
     3.Search using Interpolation Search.
     4.Exit.
     The Array must be in ascending order for using Binary and
Interpolation Searches.
     Enter Choice::4

     Program Terminated.
```

DISCUSSION:

1.The complexity of Linear Search is O(n).
2.The complexity of Binary Search is O($\log_2$ n).
3.The complexity of Interpolation Search is O(log(log n)).
4.The Interpolation and Binary Searches work on the premise that the elements are entered in ascending order.