DAA Lab-1

Name: Kshitij Kumar Sharma Roll No.: 1905514 Date: 15/07/2021

Q1)

Implement the two searching approaches (more specifically, linear search and binary search) discussed in the following. You are supposed to compare these two searching approaches based on the average number of comparisons. Execute these approaches (at least) 10 times for (at least) 50 numbers. Input: Number of elements in the array and a key to search. Output: Average number of comparisons by both approaches. For each run print, whether the key has been found or not. -- You are encouraged to use random number generator to generate the elements of the array. -- You are encouraged to use dynamically allocated arrays.

Program:

```
#include<bits/stdc++.h>
#include<stdio.h>
#include<time.h>
#include<stdlib.h>

using namespace std;

int l_search(int a[],int n,int e,int f)                    //Linear Search
        {
                int i,c1=0;

                for(i=0;i<n;i++)
        {
                c1++;
                                        //c1 is the counter which counts the no. of search
                if(e==a[i])
                {
                printf("Element found at %d position \n",i+1);
                f=1;
                break;
                }
        }
        if(f==-1)
        printf("Element not Found \n");
        return c1;
        }

int b_search(int a[],int n,int e)
        //Binary Search
        {
                int f=0,l=n-1,c2=0,m;
                while(a[m]!=e && l>=f)
        {
                c2++;
                                                // c2 is the counter which counts the no. of searches
                m=(f+l)/2;
                if(a[m]==e)
                {
                break;
                }
                if(a[m]<e)
```

```
            f=m+1;
            else
            l=m-1;
        }
        if(a[m]==e)
            printf("Element found : %d \n",a[m]);
        else
            printf("Element not found \n");
        return c2;
        }

int main()
        {
            int n,i,c1=0,c2=0,e;
            printf("Enter array size : ");
            scanf("%d",&n);
                        //Taking array size from the user
            printf("\n");

            int a[n];
            srand(time(0));
            for(int j=0;j<10;j++)
                    //Loop for comparison. 10 times in this case
            {
            for(i=0;i<n;i++)
            {
            a[i]=rand()%100000;
            //Generating random array elements
            }
            e=a[rand()%n];
                        //Selecting a random element from the array
            sort(a,a+n);
                        //Sorting the array
            c1=c1+l_search(a,n,e,-1);
            //Counting the no. of comparisons in linear search
            c2=c2+b_search(a,n,e);
            //Counting the no. of comparisons in binary search
            }
            c1=c1/10;
                                    //Average compariosns in linear search
            c2=c2/10;
                                    //Average compariosns in binary search
            printf("\n");
            printf("Average Comparisons in Linear search : %d \n",c1);
            printf("Average Comparisons in Binary search : %d \n",c2);

            return 0;
        }
```

Output:
Given in separate pdf file.