**PROGRAM1 : RECURSIVE** **Linear Search and Binary Search programs**

[1:41 pm, 01/06/2021] Muskan Gutpa: #include <stdio.h>

#include <stdlib.h>

#include <time.h>

clock\_t start, end;

double cpu\_time;

int linear\_search(int arr[], int high, int low, int key)

{

if (low<high)

return -1;

if (arr[high] == key)

return high;

if (arr[low] == key)

return low;

return linear\_search(arr,high+1,low-1,key);

}

int binary\_search(int arr[],int high, int low, int key)

{

if (low>=high)

{

int mid = (high+low)/2;

if (arr[mid]==key)

{

return mid;

}

if (arr[mid]>key)

{

return binary\_search(arr,high,mid-1,key);

}

return binary\_search(arr, mid + 1, low, key);

}

return -1;

}

int main()

{

int k,pos,c,d,i,n,temp,choice,key,j,flag=1,arr[10000];

srand(time(0));

while (flag==1)

{

printf("1:Linear\_Search\n2:Binary\_Search\n3:Exit\n");

printf("Enter your choice\n");

scanf("%d", &choice);

switch(choice)

{

case 1:

printf("Enter the number of elements:\n");

scanf("%d", &n);

for (k = 1; k <= n; k++)

{

arr[k]=rand()%100;

printf("%d ",arr[k]);

}

printf("\nEnter the Element to be Searched : \n");

scanf("%d", &key);

start = clock();

pos = linear\_search(arr, 0, n-1, key);

for (c = 1; c <= 5000; c++) for (d = 1; d <= 5000; d++) { }

end = clock();

cpu\_time = (double)(end - start) / CLOCKS\_PER\_SEC;

if(pos == -1)

{

printf("Element is not present in the Array\n");

}

else

{

printf("Element is present at the Position %d\n", pos);

}

printf("Execution time for linear\_search = %f ms\n", cpu\_time\*1000);

break;

case 2:

printf("Enter the number of elements:");

scanf("%d", &n);

for (int k =1; k<=n; k++)

{

arr[k]=rand()%100;

}

for (i=1; i <=n;i++)

{

for (j = i + 1; j <= n; ++j)

{

if (arr[i] >arr[j])

{

temp =arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

for (int k =1; k <=n; k++)

{

printf("%d ",arr[k]);

}

printf("\nEnter the element to be Searched :\n");

scanf("%d", &key);

start = clock();

for (c = 1; c <= 5000; c++) for (d = 1; d <= 5000; d++) { }

pos = binary\_search(arr, 0, n - 1, key);

end = clock();

cpu\_time = (double)(end - start) / CLOCKS\_PER\_SEC;

if(pos == -1)

{

printf("Element is not present in array\n");

}

else

{

printf("Element is present at the Position %d\n", pos);

}

printf("Execution time for binary\_search = %f ms\n", cpu\_time\*1000);

break;

default:flag=0;

}

}

return 0;

}

**OUTPUT:**



