



NATIONAL INSTITUTE OF TECHNOLOGY PUDUCHERRY

(An Institution of National Importance under MHRD, Govt. of India)

KARAIKAL – 609 609

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Roll Number: CS19B1009

Name ARUN KUMAR R

Semester: II

Class: B TECH - CSE

Subject Code: CS106

Subject Name: DATA STRUCTURES LABORATORY

1. Linear search algorithm

Date: 04/12/2019

AIM:

To write a c program to search for a key in a given array using linear search algorithm.

ALGORITHM:

1. Start the program.
2. Declare the variables.
3. Get the array and key from the user
4. Using a loop compare the key with each element present in the array and if successful return the index.
5. If key is not matched with any elements in the array, then return key not found message.
6. Output the result.
7. End the program.

PROGRAM:

```
#include<stdio.h>
int linear_search(int x[], int n, int key);
void main()
{
    int n, x[100], i, c, key;
    printf("Enter the number of elements of the array\n");
    scanf("%d", &n);

    printf("Enter the elements of the array\n");
    for(i = 0; i < n ; i++)
```

```

{
    scanf("%d",&x[i]);
}
printf("Enter the key to search\n");
scanf("%d", &key);
c = linear_search(x, n, key);
if ( c != -1)
{
    printf("The element is present in the index %d of the array\n", c);
}
else
{
    printf("The element not found\n");
}
}

int linear_search(int x[], int n, int key)
{
    int i;
    for(i = 0; i < n; i++)
    {
        if(key == x[i])
            return i;
    }
    return -1;
}

```

OUTPUT:

```

Enter the number of elements of the array
5
Enter the elements of the array
5 3 2 16 7
Enter the key to search
3
The element is present in the index 1 of the array

```

RESULT:

The program was executed successfully.

2. BINARY SEARCH

Date:04/12/2019

Aim:

To write a c program to search for a key in a given array using Binary search algorithm.

Algorithm:

1. Start the program.
2. Declare the variables.
3. Get the array and key from the user.
4. Sort the array using any algorithm.
5. Get the center element of the array and compare it with the key, if key is greater than the element, right part undergoes similar above procedure.
6. Loop or recurse until key matches and return the index.
7. Output the result.
8. End the program.

Program:

```
#include<stdio.h>
```

```
int binary_search(int x[], int l, int h, int key) // bin search using recursion function
{
    int mid;

    mid = (l+h)/2;

    if (x[mid] == key)
    {
        return mid;
    }
    if ( x[mid] < key)
    {
        return binary_search(x,mid+1,h,key);
    }
}
```

```
    else
    {
        return binary_search(x,l,mid-1,key);
    }
}
```

```
void main()
{
    int key , i, n, x[100], c;

    printf("Enter the number of elements\n");
    scanf("%d", &n);

    printf("Enter the elements\n");
    for(i = 0; i < n ; i++)
    {
        scanf("%d", &x[i]);
    }

    printf("Enter the key to find\n");
    scanf("%d", &key);

    c = binary_search(x,0,n,key);

    if ( c != -1)
    {
        printf("The element is present on the index %d of the sorted array\n ", c);
    }
    else
    {
        printf("The element is not found\n");
    }
}
```

Output:

```
Enter the number of elements
5
Enter the elements
1 2 3 4 5
Enter the key to find
3
The element is present on the index 2 of the sorted array
```

Result:

The program was executed successfully.