

Experiment1.2

Student Name: Rohit Kumar Mahato

UID: 21BCS7480

Branch: CSE

Section/Group: 717A

Semester: 3

Date of Performance:

Subject Name: Data Structure

Subject Code:21CSH-211

1. Aim:

Write a program to demonstrate linear and binary search use to find a given element in an array.

2. Objective: To know about Linear Search and Binary Search.

3. Algorithm

// Linear Search

1. Start
2. Set $i = 1$
3. If $i > n$ then go to step 7
4. If $A[i] = x$ then go to step 6
5. Set $i = i + 1$
6. Go to step 2
7. Print x found at i and go to step 8
8. Print element not found
9. exit
10. Stop

// Binary Search

1. Start
2. Bsearch(a, lb, ub, value)
3. set $beg = lb$, $end = ub$, $pos = -1$
4. repeat steps 3 and 4 while $beg \leq end$
5. set $mid = (beg + end) / 2$
6. if $a[mid] = val$
 set $pos = mid$
 print pos

- go to step 6
- else if $a[mid] > val$
- set $end = mid - 1$
- else
- set $beg = mid + 1$
- [end of if]
- [end of loop]
- 7. Step 5: if $pos = -1$
- print "value is not present in the array"
- [end of if]
- 8. exit
- 9. Stop

4. Program CODE

1.1

```
#include<iostream>
using namespace std;

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

int main(){
    int n;
    cout<<" enter the no elements";
    cin>>n;

    int arr[n];
    for(int i=0;i<n;i++){
        cin>>arr[i];
    }
    int key;
    cout<<" enter the key to find position";
    cin>>key;
```

```
    return 0;  
}
```

1.2

```
#include<iostream>  
using namespace std;  
int binarySearch(int arr[], int p, int r, int num) {  
    if (p <= r) {  
        int mid = (p + r)/2;  
        if (arr[mid] == num)  
            return mid ;  
        if (arr[mid] > num)  
            return binarySearch(arr, p, mid-1, num);  
        if (arr[mid] < num)  
            return binarySearch(arr, mid+1, r, num);  
    }  
    return -1;  
}  
int main(void) {  
    int arr[] = {1, 3, 7, 15, 18, 20, 25, 33, 36, 40};  
    int n = sizeof(arr)/ sizeof(arr[0]);  
    int num;  
    cout << "Enter the number to search: \n";  
    cin >> num;  
    int index = binarySearch (arr, 0, n-1, num);  
    if(index == -1){  
        cout<< num <<" is not present in the array";  
    }else{  
        cout<< num <<" is present at index "<< index <<" in the array";  
    }  
    return 0;  
}
```

Output

1.1

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\mahat\OneDrive\Desktop\DATA STRUCTURE LAB> cd "c:\Users\mahat\OneDrive\Desktop\DATA STRUCTURE LAB\" ; if ($?) { g++ 04_searching_element.cpp -o 04_searching_element } ; if ($?) { .\04_searching_element }
enter the no elements 5
10 12 51 20 13
enter the key to find position 51
2
PS C:\Users\mahat\OneDrive\Desktop\DATA STRUCTURE LAB> 
```

1.2

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\mahat\OneDrive\Desktop\DATA STRUCTURE LAB> cd "c:\Users\mahat\OneDrive\Desktop\DATA STRUCTURE LAB\" ; if ($?) { g++ 05_binary_code.cpp -o 05_binary_code } ; if ($?) { .\05_binary_code }
Enter the number to search:
20
20 is present at index 5 in the array
PS C:\Users\mahat\OneDrive\Desktop\DATA STRUCTURE LAB> 
```

Learning Outcomes:

1. To use linear and binary search