

Practical # 10

Objective:

*Discuss binary search algorithm in data structure. **Design** a C++ program for binary search algorithm.*

Theory:

In this Lab, we discuss the binary search technique. Binary search algorithm finds a given element in a list of elements with **$O(\log n)$** time complexity where **n** is total number of elements in the list. The binary search algorithm can be used with only a sorted list of elements. That means the binary search is used only with a list of elements that are already arranged in an order. This search process starts comparing the search element with the middle element in the list. If both are matched, then the result is "element found". Otherwise, we check whether the search element is smaller or larger than the middle element in the list. If the search element is smaller, then we repeat the same process for the left sublist of the middle element. If the search element is larger, then we repeat the same process for the right sublist of the middle element. We repeat this process until we find the search element in the list or until we left with a sublist of only one element. And if that element also doesn't match with the search element, then the result is "Element not found in the list".

Binary search is implemented using following steps:

Step 1 - Read the search element from the user.

Step 2 - Find the middle element in the sorted list.

Step 3 - Compare the search element with the middle element in the sorted list.

Step 4 - If both are matched, then display "Given element is found!!!" and terminate the function.

Step 5 - If both are not matched, then check whether the search element is smaller or larger than the middle element.

Step 6 - If the search element is smaller than middle element, repeat steps 2, 3, 4 and 5 for the left sublist of the middle element.

Step 7 - If the search element is larger than middle element, repeat steps 2, 3, 4 and 5 for the right sublist of the middle element.

Step 8 - Repeat the same process until we find the search element in the list or until sublist contains only one element.

Step 9 - If that element also doesn't match with the search element, then display "Element is not found in the list!!!" and terminate the function.

Lab Objectives:

- To be able to write C++ program for binary search algorithm.

C++ program: Write C++ program for binary search algorithm.

```
#include<iostream>
using namespace std;
int main()
{
    int i, arr[10], num, first, last, middle;
    cout<<"Enter 10 Elements (in ascending order): ";
    for(i=0; i<10; i++)
        cin>>arr[i];
    cout<<"\nEnter Element to be Search: ";
    cin>>num;
    first = 0;
    last = 9;
    middle = (first+last)/2;
    while(first <= last)
    {
        if(arr[middle]<num)
            first = middle+1;
        else if(arr[middle]==num)
        {
            cout<<"\nThe number, "<<num<<" found at Position "<<middle+1;
            break;
        }
        else
            last = middle-1;
        middle = (first+last)/2;
    }
    if(first>last)
        cout<<"\nThe number, "<<num<<" is not found in given Array";
    cout<<endl;
    return 0;
}
```

OUTPUT

```
Enter 10 Elements (in ascending order): 12
13
14
15
16
17
18
19
20
21

Enter Element to be Search: 17

The number, 17 found at Position 6
```

Review Questions/ Exercise:

1. Implement binary search algorithm using function in C++.

2. Implement binary search algorithm that allow user to define size of array in C++.

Name: _____

Roll #: _____

Date: _____

Remarks:

Subject Teacher