Name : Pratik Rajesh Jade

Roll no : A 70

Aim - Implement Binary search program with Divide and Conquer design strategy for n numbers using C++. Discuss Best, Average and Worst time complexity.

*//Recursive implementation of Binary Search:*

#include <iostream>

using namespace std;

int binarySearch(int arr[], int l, int r, int x) {

    if (r >= l) {

        int mid = l + (r - l) / 2;

        if (arr[mid] == x)

            return mid;

        if (arr[mid] > x)

            return binarySearch(arr, l, mid - 1, x);

        return binarySearch(arr, mid + 1, r, x);

    }

    return -1;

}

int main() {

    int arr[] = {2, 3, 4, 10, 40};

    int n = sizeof(arr) / sizeof(arr[0]);

    int x;

    cout << "Enter the number to be searched: ";

    cin >> x;

    int result = binarySearch(arr, 0, n - 1, x);

    if (result == -1)

        cout << "Element not present in the array";

    else

        cout << "Element found at index " << result << endl;

    return 0;

}

*// Best time complexity: O(1)*

*// Average time complexity: O(log n)*

*// Worst time complexity: O(log n)*

Output:

