

## DESIGN AND ANALYSIS OF ALGORITHMS LAB ( PCS-409 )

### WEEK 1

1 . Problem Statement: Given an array of nonnegative integers, design a linear algorithm and implement it using a program to find whether given key element is present in the array or not. Also, find total number of comparisons for each input case. (Time Complexity =  $O(n)$  )

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

// Function to perform linear search
void linearSearch(const vector<int>& arr, int key, ofstream& fout) {
    int n = arr.size();
    int comparisons = 0;
    int i = 0;
    while (i < n) {
        comparisons++;

        // Count each comparison made
        if (arr[i] == key) {
            fout << "Element found. ";
            break;
        }
        i++;
    }
    if (i == n) {
        // Worst case: Element not found
        fout << "Element not found. ";
    }
    fout << "Number of comparisons done: " << comparisons << endl;
}
```

## DESIGN AND ANALYSIS OF ALGORITHMS LAB ( PCS-409 )

```
int main() {

                                                                    // Open input and output files

    ifstream fin("input.txt");
    ofstream fout("output.txt");

                                                                    // Check if files are opened successfully

    if (!fin.is_open() || !fout.is_open()) {
        cout << "Error occurred while opening files.\n";
        return 0;
    }
    int t;

                                                                    // Number of test cases

    fin >> t;
    while (t-- > 0) {
        int n;
        fin >> n;
        vector<int> arr(n); // Declare array of size n

                                                                    // Input array elements

        for (int i = 0; i < n; i++) {
            fin >> arr[i];
        }

        int key;

                                                                    // Key to be searched

        fin >> key;

                                                                    // Perform linear search and output the result

        linearSearch(arr, key, fout);
    }

                                                                    // Close input and output files

    fin.close();
    fout.close();
}
```

## DESIGN AND ANALYSIS OF ALGORITHMS LAB ( PCS-409 )

main.cpp	input.txt	:	output.txt	:
1	5			
2	8			
3	34 35 65 31 25 89 64 30			
4	89			
5	5			
6	977 354 244 546 355			
7	244			
8	6			
9	23 64 13 67 43 56			
10	63			
11	10			
12	1 2 3 4 5 6 7 8 9 10			
13	10			
14	100			
15	6 78 36 100 95 12 22 91 69 88 60 83 56 77 1			
16	48 46 39 40 41 31 14 33 35 71 27 30 58 89 57			
17	10 84 32 47 65 25 64 68 98 44 74 99 26 3 76			
18	96 70 92 73 53 51 49 29 52 20 63 24 21 61 38			
19	5 94 97 67 86 75 37 34 11 7 79 50 80 62 19 28			
20	90 59 4 18 16 8 45 93 82 54 66 85 81 42 9 15			
21	43 72 17 87 55 23 13 2			
22	61			

## DESIGN AND ANALYSIS OF ALGORITHMS LAB ( PCS-409 )

main.cpp	input.txt	:	output.txt	:
1	Element found. Number of comparisons done: 6			
2	Element found. Number of comparisons done: 3			
3	Element not found. Number of comparisons done: 6			
4	Element found. Number of comparisons done: 10			
5	Element found. Number of comparisons done: 59			
6				

## DESIGN AND ANALYSIS OF ALGORITHMS LAB ( PCS-409 )

2 . Problem Statement: Given an already sorted array of positive integers, design an algorithm and implement it using a program to find whether given key element is present in the array or not. Also, find total number of comparisons for each input case. (Time Complexity =  $O(n \log n)$ , where  $n$  is the size of input)

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

// Function to perform binary search
void BinarySearch(const vector<int>& arr, int key, ofstream& fout) {
    int low =0,high=arr.size()-1;
    int comparisons=0;
    while (low<=high) {
        comparisons++; // Count each comparison made
        int mid = (low+high)/2;
        if (arr[mid]==key) {
            fout << "Element found. ";
            break; //break if element is found
        }
        else if(arr[mid]>key){
            high = mid-1;
        }
        else{
            low = mid+1;
        }
    }
    if (high<low) { //worst case
        fout << "Element not found. ";
    }
    fout << "Number of comparisons done: " << comparisons << endl;
}
```

## DESIGN AND ANALYSIS OF ALGORITHMS LAB ( PCS-409 )

```
int main() {
    ifstream fin("input.txt");
    ofstream fout("output.txt");

    if (!fin.is_open() || !fout.is_open()) {
        cout << "Error occurred while opening files.\n";
        return 0;
    }

    int t;                                     // Number of test cases
    fin >> t;

    while (t-- > 0) {
        int n;
        fin >> n;
        vector<int> arr(n);                  // Declare array of size n

        for (int i = 0; i < n; i++) {
            fin >> arr[i];                   // Input array elements
        }

        int key;
        fin >> key;                          // Input the key to be searched

        BinarySearch(arr, key, fout);        // Binary Search
    }

    fin.close();
    fout.close();
}
```

## DESIGN AND ANALYSIS OF ALGORITHMS LAB ( PCS-409 )

```
main.cpp  input.txt  ⋮  output.txt  ⋮
1  5
2  5
3  12 23 36 39 41
4  41
5  8
6  21 39 40 45 51 54 68 72
7  69
8  10
9  101 246 438 561 796 896 899 4644 7999 8545
10 7999
11 10
12 1 2 3 4 5 6 7 8 9 10
13 10
14 100
15 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16 16 17 18 19 20 21 22 23 24 25 26 27
17 28 29 30 31 32 33 34 35 36 37 38 39
18 40 41 42 43 44 45 46 47 48 49 50 51
19 52 53 54 55 56 57 58 59 60 61 62 63
20 64 65 66 67 68 69 70 71 72 73 74 75
21 76 77 78 79 80 81 82 83 84 85 86 87
22 88 89 90 91 92 93 94 95 96 97 98 99 100
23 61
```

## DESIGN AND ANALYSIS OF ALGORITHMS LAB ( PCS-409 )

main.cpp	input.txt	:	output.txt	:
1	Element found. Number of comparisons done: 3			
2	Element not found. Number of comparisons done: 4			
3	Element found. Number of comparisons done: 3			
4	Element found. Number of comparisons done: 4			
5	Element found. Number of comparisons done: 7			
6				



## DESIGN AND ANALYSIS OF ALGORITHMS LAB ( PCS-409 )

3 . Problem Statement: Given an already sorted array of positive integers, design an algorithm and implement it using a program to find whether a given key element is present in the sorted array or not. For an array  $arr[n]$ , search at the indexes  $arr[0]$ ,  $arr[2]$ ,  $arr[4]$ ,....., $arr[2k]$  and so on. Once the interval  $(arr[2k] < key < arr[2k+1])$  is found, perform a linear search operation from the index  $2k$  to find the element key. (Complexity  $< O(n)$ , where  $n$  is the number of elements need to be scanned for searching):

```
#include <iostream>
#include<vector>
#include <fstream>
#include<cmath>
using namespace std;

// Function to perform jump search
void jumpSearch(vector<int>arr, int key,ofstream &fout) {
    int n = arr.size();
    int start = 0, comp1 = 0, flag = 0;
    int end = sqrt(n);
    while(arr[end] <= key && end < n) {
        comp1++;
        start = end;
        end += sqrt(n);
        if(end > n-1) end = n;
    }
    int comp2=0;
    for(int i=start;i<end;i++) {
        comp2++;
        if(arr[i] == key) {
            flag = true;
            break;
        }
    }

    if(flag) fout<<"Present ";
```

## DESIGN AND ANALYSIS OF ALGORITHMS LAB ( PCS-409 )

```
else fout<<"Not present ";
fout<<comp1<<"+"<<comp2;
fout<<endl;

}

int main() {
    ifstream fin("input.txt");
    ofstream fout("output.txt");

    if (!fin.is_open() || !fout.is_open()) {
        cout << "Error occurred while opening files.\n";
        return 0;
    }

    int t; // Number of test cases
    fin >> t;

    while (t--> 0) {
        int n; // Array size
        fin >> n;
        vector<int>arr(n); // Declare array of size n

        for (int i = 0; i < n; i++) {
            fin >> arr[i]; // Input array elements
        }
        int key;
        fin >> key; // Input the key to be searched

        jumpSearch(arr, key,fout); //jump Search
    }
    fin.close();
    fout.close();
}
```

## DESIGN AND ANALYSIS OF ALGORITHMS LAB ( PCS-409 )

```
main.cpp  input.txt  ⋮  output.txt  ⋮
1  5
2  5
3  12 23 36 39 41
4  41
5  8
6  21 39 40 45 51 54 68 72
7  69
8  10
9  101 246 438 561 796 896 899 4644 7999 8545
10 7999
11 10
12 1 2 3 4 5 6 7 8 9 10
13 10
14 100
15 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16 16 17 18 19 20 21 22 23 24 25 26 27
17 28 29 30 31 32 33 34 35 36 37 38 39
18 40 41 42 43 44 45 46 47 48 49 50 51
19 52 53 54 55 56 57 58 59 60 61 62 63
20 64 65 66 67 68 69 70 71 72 73 74 75
21 76 77 78 79 80 81 82 83 84 85 86 87
22 88 89 90 91 92 93 94 95 96 97 98 99 100
23 61
```

## DESIGN AND ANALYSIS OF ALGORITHMS LAB ( PCS-409 )

main.cpp	input.txt	output.txt
1	Present 2+1	
2	Not present 3+2	
3	Present 2+3	
4	Present 3+1	
5	Present 6+1	
6		