



# CHITTAGONG UNIVERSITY OF ENGINEERING AND TECHNOLOGY

## Department of Computer Science & Engineering

**Course No: CSE- 244**

**Course Title:**  
**Algorithm Design and Analysis (Sessional)**

**Experiment No: 01**

**Name Of the Experiment: Introduction to algorithm**

### Identity Details

**Name of the Student: Md. Safayat Bin Nasir**

**Student Id Number: 2004099**

**Level: 2 Term: 2 Section: B**

**Group: B1**

***Date of performance: 20-08-2023***

***Date of Submission: 03-09-2023***

**Remarks**

**Program 1:** Given a list of numbers , sort them in ascending order using bubble sort algorithm. After implementing the program analyze its complexity.

---

```
#include <bits/stdc++.h>
using namespace std;
vector<int> bubble(vector<int>&v){
    for(int i = 0 ; i < v.size() ; i++){
        for(int j = 0 ; j < v.size(); j++){
            if(v[i] <= v[j]){
                int temp = v[j];
                v[j] = v[i];
                v[i] = temp;
            }
        }
    }
    return v;
}
int main()
{
    int n;
    cin >> n;
    vector<int>v(n);
    for(auto &x:v) cin >> x;
    v = bubble(v);
    for(int i = 0 ; i < n ;i++){
        cout << v[i] << " ";
    }
    cout << endl;
}
```

**Output:**

```
Input:
5
9 31 4 12 -4

Expected Output:
-4 4 9 12 31

Received Output:
-4 4 9 12 31
```

**Program 2:** Given a list of sorted numbers in ascending order, find a specific element from that list using binary search. After implementing the program analyze its complexity.

---

```
#include<bits/stdc++.h>
using namespace std;
int main(){
    int n ;
    cin >> n;
    vector<int>v(n);
    for(auto &x:v) cin >> x;
    int x;
    cin >> x;
    int l = 0, r = n , ind = - 1;
    while(l <= r){
        int mid = l + (r-l) /2;
        if(v[mid] == x){
            ind = mid;
            break;
        }
        if(v[mid] > x) r = mid-1;
        else l = mid+1;
    }
    if(ind == -1){
        cout << x << " is not found." << endl;
    }
    else{
        cout << x << " is found at index " << ind << endl;
    }
}
```

## Output:

^ Testcase 1 Passed 25ms

Input:

5

-4 4 9 12 31

12

Expected Output:

12 is found at index 3

Received Output:

12 is found at index 3

^ Testcase 2 Passed 36ms

Input:

5

-4 4 9 12 31

11

Expected Output:

11 is not found.

Received Output:

11 is not found.

**Program 3:** Given a list of sorted numbers in ascending order, find how many times a specific element is presented in that list. Complexity of the code must be  $O(\log n)$ .

---

```
#include<bits/stdc++.h>
using namespace std;
int main(){
    int n , m;
    cin >> n;
    vector<int>v(n);
    for(auto &x:v) cin >> x;
    cin >> m;
    int l = 0, r = n ,ft = - 1 , last = -1;
    while(l <= r){
        int mid = l + (r-l) /2;
        if(v[mid] == m){
            ft = mid;
            r = mid - 1;
        }
        if(v[mid] > m) r = mid-1;
        else l = mid+1;
    }
    l = 0 , r = n;
    while(l <= r){
        int mid = l + (r-l) /2;
        if(v[mid] == m) last = mid,l = mid + 1;
        if(v[mid] > m) r = mid-1;
        else l = mid+1;
    }
    if(ft == -1 and last == - 1) cout << m << " is absent." << endl;
    else cout << m << " is presented " << last - ft + 1 << " times"<< endl;
}
```

**Output:**

Input:

5

-4 4 9 9 31

9

Expected Output:

9 is presented 2 times

Received Output:

9 is presented 2 times

