



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

## Experiment – 3

**Student Name:** Himanshu Gupta

**Branch:** BE-CSE

**Semester:** 5<sup>th</sup>

**UID:** 23BCS10889

**Section/Group:** KRG-2B

**Date of Performance:** 7/8/25

**Subject Name:** Design and Analysis of Algorithms

**Subject Code:** 23CSH-301

**1. Aim:** To find frequency of an element in a given array in  $O(1)$  time complexity.

**2. Objective:** The main objective is to efficiently determine the frequency of each element in an array using **HashMap** (hashing technique) to reduce time complexity compared to nested loops.

### **3. Input/ Apparatus Used:**

A HashMap(or dictionary) is used to store array elements as keys and their frequencies as values.

### **4. Algorithm:**

**Naive Algorithm (using array traversal –  $O(N^2)$ ):**

1. Input the number of elements in the array and then array elements.
2. For each element, traverse the array to count its frequency.
3. Print each element with its frequency.

This takes  $O(N^2)$  in the worst case.

### **Optimized Approach(using Hashing- $O(N)$ -**

1. Input the number of elements in an array.
2. Input the array elements.
3. Create a HashMap(key=element, value= frequency).
4. Traverse the array:
  - For each element, increase its count in the HashMap.
5. Traverse the HashMap and print each element with its frequency.



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

## 5. Step- wise Pseudocode/Algorithm used-

function findFrequency(arr, n):

    create empty HashMap hm

    for i = 0 to n-1:

        if arr[i] exists in hm:

            hm[arr[i]] = hm[arr[i]] + 1

        else:

            hm[arr[i]] = 1

    for each key in hm:

        print key, hm[key]

## 6. Code and output:

```
class Solution {  
    public ArrayList<ArrayList<Integer>> countFreq(int[] arr) {  
        // code here  
        ArrayList<ArrayList<Integer>> ans = new ArrayList<>();  
        HashMap<Integer, Integer> map = new HashMap<>();  
        for(int i=0; i<arr.length; i++){  
            if(map.containsKey(arr[i])) map.put(arr[i],map.get(arr[i])+1);  
            else map.put(arr[i],1);  
        }  
  
        for(int key : map.keySet()){  
            ArrayList<Integer> list = new ArrayList<>();  
            list.add(key); list.add(map.get(key));  
            ans.add(list);  
        }  
        return ans;  
    }  
}
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

The screenshot shows a programming competition interface. At the top, there are navigation links: Courses, Tutorials, Practice, and Jobs. On the right, there's a search bar, a timer icon (36), and other user-related icons. The main area is divided into sections: Problem, Editorial, Submissions, and Comments. The Problem section shows the code submitted by the user:

```
1* class Solution {  
2*     public ArrayList<ArrayList<Integer>> countFreq(int[] arr) {  
3*         // code here  
4*         ArrayList<ArrayList<Integer>> ans = new ArrayList<>();  
5*         HashMap<Integer, Integer> map = new HashMap<>();  
6*         for(int i=0; i<arr.length; i++){  
7*             if(map.containsKey(arr[i])) map.put(arr[i],map.get(arr[i])+1);  
8*             else map.put(arr[i],1);  
9*         }  
10*        for(int key : map.keySet()){  
11*            ArrayList<Integer> list = new ArrayList<>();  
12*            list.add(key); list.add(map.get(key));  
13*            ans.add(list);  
14*        }  
15*    }  
16*    return ans;  
17* }  
18* }  
19* }
```

Below the code, the message "Problem Solved Successfully" is displayed with a green checkmark. There are also "Suggest Feedback" and "Custom Input" buttons. To the left, the "Output Window" shows "Test Cases Passed: 1112 / 1112" and "Attempts: Correct / Total: 1 / 1" with an accuracy of 100%. Below these, "Points Scored: 2 / 2" and "Time Taken: 2.48" are shown. A note says "Your Total Score: 92". At the bottom, there are buttons for "Custom Input", "Compile & Run", and "Submit".