



Experiment – 3

Student Name: Himanshu Gupta

UID: 23BCS10889

Branch: BE-CSE

Section/Group: KRG-2B

Semester: 5th

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.

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.

Courses ▾ Tutorials ▾ Practice ▾ Jobs ▾



Search...



Problem

Editorial

Submissions


Comments

Output Window

Compilation Results

Custom Input

Y.O.G.I. (AI Bot)

Problem Solved Successfully 

Suggest Feedback


Test Cases Passed

1112 / 1112


Attempts: Correct / Total

1 / 1

Accuracy: 100%

Points Scored 

2 / 2

Your Total Score: 92 

Time Taken

2.48





Solve Next

Largest subarray of 0's and 1's


Largest subarray with 0 sum

Java (21)

Start Timer



```
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
11         for(int key : map.keySet()){
12             ArrayList<Integer> list = new ArrayList<>();
13             list.add(key); list.add(map.get(key));
14             ans.add(list);
15         }
16
17         return ans;
18     }
19 }
```



Custom Input

Compile & Run

Submit