DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

NAAC GRADE A+
Accredited University

UNIVERSITY INSTITUTE OF ENGINEERING

Department of Computer Science & Engineering

(BE-CSE/IT-5th Sem)

Design and Analysis of Algorithms

Subject Code: 23CSH-301/ITH-301

Submitted to:

Faculty name: MD. Shaqlain

Submitted by:

Name: HarshVardhan

UID: 23BCS10363
Section: Krg-1

Group: A

Name: Aaditya Bansal **UID: 23BCS11115**

INDEX

Ex. No	List of Experiments	Date	Conduct (MM: 12)	Viva (MM: 10)	Worksheet (MM: 8)	Total (MM:30)	Remarks/Signature
1.1	Analyze if stack Isempty, Isfull and if elements are present then return top element in stacks using templates and also perform push and pop operation in stack.	21/07/25					
1.2	Develop a program for implementation of power function and determine that complexity should be O (log n).	01/08/25					
1.3	Evaluate the complexity of the developed program to find frequency of elements in a given array.	08/08/25					
1.4	i. Apply the concept of Linked list and write code to Insert and Delete an element at the beginning and end of Singly Linked List. ii. Apply the concept of Linked list and write code to Insert and Delete an element at the beginning and at end in Doubly and Circular Linked List.						
2.1	Sort a given set of elements using the Quick sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be						

	sorted. The elements can be read from a file or can be generated using the random number generator.						
2.2	Develop a program and analyze complexity to implement subset-sum problem using Dynamic Programming.						
2.3	Develop a program and analyze complexity to implement 0-1 Knapsack using Dynamic Programming.						
3.1	Develop a program and analyze complexity to find shortest paths in a graph with positive edge weights using Dijkstra's algorithm.						
3.2	Develop a program and analyze complexity to find all occurrences of a pattern P in a given string S.						
3.3	Lab Based Mini Project.						

Experiment No: 1.3

Student Name: HarshVardhan
UID: 23BCS10363
Branch: CSE
Section/Group: Krg-1A
Semester: 5th
Date of Performance: 08/08/25
Subject Name: Design analysis and algorithm
Subject Code: 23CSH-301

Aim:

Code to find frequency of elements in a given array in O(N) time complexity.

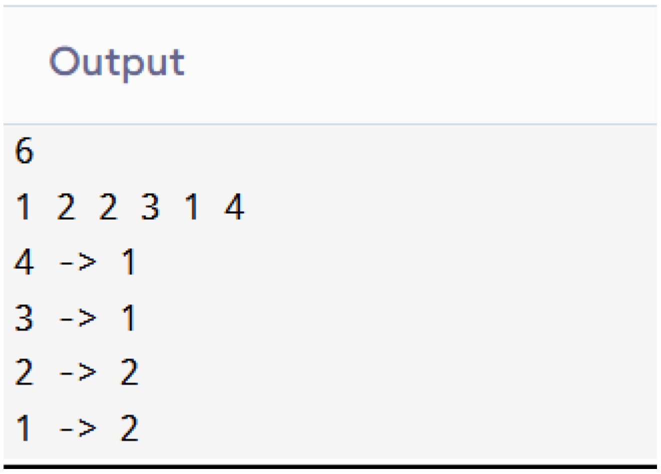
Procedure:

1. Input:
   - An array arr of size n.
2. Initialize:
   - A hash map (dictionary) freq to store element → frequency mapping.
3. Traverse the array (from index 0 to n-1):
   - For each element arr[i]:
     - → If arr[i] is not in freq, insert it with value 1.
     - → Else, increment freq[arr[i]] by 1.
4. After traversal:
   - The hash map freq will contain each unique element and its frequency.
5. Output:
   - For each key-value pair in freq, print element -> frequency.

Code:

```
#include <bits/stdc++.h>
using namespace std;
int main() {
    int n;
    cin >> n;  // size of array
    vector<int> arr(n);
for (int i = 0; i < n; i++) {
        cin >> arr[i];}
      unordered_map<int, int> freq;
   for (int i = 0; i < n; i++) {
        freq[arr[i]]++;}
   for (auto &it : freq) {
cout << it.first << " -> " << it.second<< endl; }
      return 0;
   }
```

Output:



```
6
1 2 2 3 1 4
4 -> 1
3 -> 1
2 -> 2
1 -> 2
```

Time Complexity:

- Reading input: O(n)
- Counting frequencies: Each insertion/update in an unordered_map is on average O(1), so total is O(n).
- Printing the frequencies: The number of unique elements is at most n, so in the worst case, O(n).

Overall, the total time complexity is O(n) on average, considering the hash map operations are constant time on average.

Learning outcomes:

1. Understand the concept of arrays and hash maps (unordered_map).
2. Learn how to store and retrieve key–value pairs efficiently.
3. Recognize the use of hashing to achieve O(n)O(n)O(n) time complexity.