Experiment 1.3

Aim: Evaluate the complexity of the developed program to find frequency of elements in a given array

Objectives: To implement power function in O(n) time complexity.

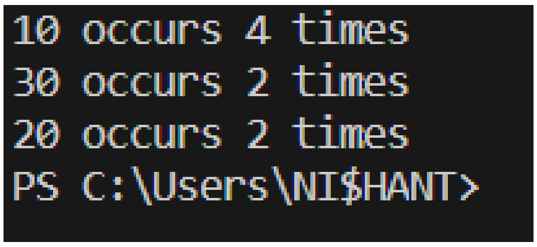
Input/Apparatus Used: In this program, HashMap concept is used in order to get less complexity.

Procedure/Algorithm:

a) Initialize an empty hash map to store the frequency of elements.
b) Iterate through the given array.
For each element:
•Check if the element exists in the hash map.
• If it does, increment its corresponding frequency.
• If it doesn't, add the element to the hash map with a frequency of 1.
c) After iterating through the entire array, the hash map will contain frequencies of all elements.
d) Iterate through the hash map to print or store the frequencies of elements.
Sample Code:

```
import java.util.*;

public class DAAexp3 {
    public static void countFreq(int arr[], int n) {
        boolean visited[] = new boolean[n];
        Arrays.fill(visited, false);

        // Traverse through array elements and
        // count frequencies
        for (int i = 0; i < n; i++) {

            // Skip this element if already processed
            if (visited[i] == true)
                continue;

            // Count frequency
            int count = 1;
            for (int j = i + 1; j < n; j++) {
                if (arr[i] == arr[j]) {
                    visited[j] = true;
                    count++;
                }
            }
            System.out.println(arr[i] + " occurs " + count + " times ");
        }
    }

    public static void main(String[] args) {
        int arr[] = new int[] { 10, 30, 10, 20, 10, 20, 30, 10 };
        int n = arr.length;
        countFreq(arr, n);
    }
}
```

Observations/Outcome :



```
10 occurs 4 times
30 occurs 2 times
20 occurs 2 times
PS C:\Users\NI$HANT>
```

Time Complexity: O(n)