EXPERIMENT NO: 8

AIM: implement FM algorithm using python / R

Theory:

The Flajolet-Martin (FM) algorithm is a probabilistic algorithm used to estimate the number of distinct elements in a stream of data without storing or revisiting each data item. This algorithm is particularly useful in big data analytics and data stream processing, where it's often infeasible to store and process every data point individually.

Key Features and Concepts:

1. **Probabilistic Counting**: The FM algorithm uses hash functions and probability theory to provide an approximate count of distinct elements. It leverages the principle that different hash functions produce distinct values with a certain probability.
2. **Bit Patterns**: Instead of counting distinct elements directly, the FM algorithm focuses on the leading zeros in the binary representation of the hash values of the data elements. These patterns are used to estimate the number of distinct elements.
3. **Randomization**: The algorithm introduces randomization by using multiple hash functions. Each hash function is applied to each data element independently, and the maximum number of leading zeros across all hash values is tracked.
4. **Estimation Process**: The FM algorithm calculates the estimate of the number of distinct elements using the formula 2^(maximum leading zeros). This estimation provides a probabilistic count, which may have some error but is highly memory-efficient.

Use Cases:

1. **Large Data Streams**: FM is particularly useful in scenarios where the data stream is too large to store and process in memory. It provides a memory-efficient method for estimating distinct element counts.
2. **Web Traffic Analysis**: In web analytics, the FM algorithm can estimate the number of distinct visitors or IP addresses accessing a website in real-time.

CODE:

```
import java.util.Scanner;

public class Main {

    static int hash(int x) {
        return (6 * x + 1) % 5;
    }

    static String toThreeBitBinary(int num) {
        String binary = Integer.toBinaryString(num);
        while (binary.length() < 3) {
            binary = "0" + binary;
        }
        return binary;
    }
    static void countTrailingZeros(String[] arr, int[] result) {
        for (int i = 0; i < arr.length; i++) {
            String binary = arr[i];
            int count = 0;
            boolean encounteredOne = false;
            for (int j = binary.length() - 1; j >= 0; j--) {
                if (binary.charAt(j) == '0' && !encounteredOne) {
                    count++;
                } else if (binary.charAt(j) == '1') {
                    encounteredOne = true;
                }
            }
            if (!encounteredOne) {
                count = 0;
            }
            result[i] = count;
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the size of the array: ");
        int size = scanner.nextInt();
        int[] inputArray = new int[size];

        System.out.println("Enter elements of the array:");
        for (int i = 0; i < size; i++) {
            System.out.print("Element " + (i + 1) + ": ");
            inputArray[i] = scanner.nextInt();
        }

        int[] hashedArray = new int[size];
        for (int i = 0; i < size; i++) {
            hashedArray[i] = hash(inputArray[i]);
        }
```

```
        String[] binaryArray = new String[size];
        for (int i = 0; i < size; i++) {
            binaryArray[i] = toThreeBitBinary(hashedArray[i]);
        }
        hint[] trailingZerosArray = new int[size];
        countTrailingZeros(binaryArray, trailingZerosArray);

        int maxTrailingZeros = Integer.MIN_VALUE;
        for (int count : trailingZerosArray) {
            if (count > maxTrailingZeros) {
                maxTrailingZeros = count;
            }
        }

        System.out.println("Maximum trailing zeros: " + maxTrailingZeros);
        double powerOfTwo = Math.pow(2, maxTrailingZeros);
        System.out.println("the number of distinct elements " +
powerOfTwo);

        scanner.close();
    }
}
```

OUTPUT:

```
"C:\Program Files\Java\jdk-20\bin\java.exe" "-ja
Enter the size of the array: 12
Enter elements of the array:
Element 1: 1
Element 2: 3
Element 3: 2
Element 4: 1
Element 5: 2
Element 6: 3
Element 7: 4
Element 8: 3
Element 9: 1
Element 10: 2
Element 11: 3
Element 12: 1
Maximum trailing zeros: 2
the number of distinct elements 4.0
```