AY: 2025-26

Class:		**Semester:**	
Course Code:		**Course Name:**	

Name of Student:	BARI ANKIT VINOD
Roll No. :	61
Experiment No.:	**7**
Title of the Experiment:	**To implement the Flajolet-Martin algorithm for estimating the number of distinct elements**
Date of Performance:	
Date of Submission:	

Evaluation

Performance Indicator	Max. Marks	Marks Obtained
Performance	5	
Understanding	5	
Journal work and timely submission	10	
Total	20	

Performance Indicator	Exceed Expectations (EE)	Meet Expectations (ME)	Below Expectations(BE)
Performance	4-5	2-3	1
Understanding	4-5	2-3	1
Journal work and timely submission	8-10	5-8	1-4

Checked by

Name of Faculty :

Signature :

Date :

Experiment No.: 7
AIM:

To implement the Flajolet-Martin algorithm for estimating the number of distinct elements in a data stream using any programming language.

THEORY:

The Flajolet-Martin algorithm is a probabilistic algorithm used for estimating the number of distinct elements (cardinality) in a large data stream.
Instead of storing all elements (which is memory-expensive), it uses hashing and bit-pattern observations to provide an approximate distinct count efficiently.

Concept:

Each element in the stream is hashed to a binary value.

The algorithm records the position of the rightmost 1-bit in each hash value.

The position of the rightmost 1-bit gives a measure of how often that bit pattern appears, helping estimate the cardinality.

The final estimation is made using the formula:

Estimated Count
=
2
R
/
ϕ
Estimated Count=2
R
/ϕ

where
R
R = position of the rightmost 1-bit,
and
ϕ
$\approx$
0.77351
ϕ≈0.77351 (correction constant).

This algorithm is widely used in Big Data and streaming analytics for approximate distinct counts — such as counting unique website visitors or hashtags on social media streams.

IMPLEMENTATION (Python Example):

```
import hashlib
import math
```

```
# Function to get position of rightmost 1-bit
def rho(x):
    return len(x) - x.rfind('1')

# Hash function to convert data to binary string
def hash_func(value):
    hash_val = hashlib.md5(value.encode()).hexdigest()
    bin_val = bin(int(hash_val, 16))[2:]
    return bin_val

# Flajolet-Martin Algorithm
def flajolet_martin(stream):
    max_rho = 0
    for item in stream:
        h = hash_func(item)
        r = rho(h)
        if r > max_rho:
            max_rho = r
    phi = 0.77351
    return int((2 ** max_rho) / phi)

# Example Stream
stream_data = ["apple", "banana", "orange", "apple", "grape", "banana", "pear"]
estimate = flajolet_martin(stream_data)
print("Estimated number of distinct elements:", estimate)
```

STEPS TO EXECUTE:

Install Python 3.12 or later.

Open any Python IDE (VS Code, Anaconda, Jupyter Notebook).

Copy the above code into a new file named flajolet_martin.py.

Run the program using:

```
python flajolet_martin.py
```

Observe the estimated count of unique elements in the stream.

CONCLUSION:

The Flajolet-Martin algorithm was successfully implemented.
It efficiently estimated the number of distinct elements in a given data stream without storing all data points, demonstrating its usefulness for large-scale streaming analytics.