Terna Engineering College
Computer Engineering Department

Program: Sem VII

Course: Big Data Analytics & Computational Lab -I (BDA&CL-I)

Experiment No. 08

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per the following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case there is no Blackboard access available)

Roll No. 50	Name: AMEY THAKUR
Class: BE-COMPS-50	Batch: B3
Date of Experiment: 05-10-2021	Date of Submission: 05-10-2021
Grade :	

Aim: To implement the DGIM algorithm.

B.1 Software code written by a student:

```
inp = list(map(int,input("Enter Elements : ").split()))
bucket_list = []
bucket_size_count = {}

def checker():
    for ct in bucket_size_count.keys():
        if(bucket_size_count[ct]>2):
            s2,e2,size2 = bucket_list.pop(-2)
            s1,e1,size1 = bucket_list.pop(-2)
            bucket_list.insert(-1,(s1,e2,size1*2))
            bucket_size_count[ct]-=2

start_index =0
end_index = 0
pair = 0

for i in range(len(inp)):
    bit = inp[i]
    if(bit == 1):
        if(pair == 1):
```

```
            end_index = i
            pair = 0
            bucket_list.append((start_index,end_index,2))
            if 2 in bucket_size_count:
                bucket_size_count[2]+=1
            else:
                bucket_size_count[2] = 1
            checker()
        else:
            start_index = i
            pair = 1


print ("Bucket Indexes Are : ",bucket_list)
starts = []
ends = []
for s,e,size in bucket_list:
    starts.append(s)
    ends.append(e)

print ("Buckets are ",end="")

for i in range(len(inp)):
    bit = inp[i]
    if(i in starts):
        print (" " ,bit,end="")
    elif(i in ends):
        print (bit,end= " ")
    else:
        print (bit,end = " ")

k = int(input("\nEnter window size : "))
length = len(inp)

bound1 = length-1-k
bound2 = length-1

ones_count = 0

for s,e,size in bucket_list[::-1]:
    if(s<bound1 and e < bound1):
        break
    elif(s<=bound1 and e >= bound1):
```

```
        ones_count +=int(size/2)
    elif(s>=bound1 and e >= bound1):
        ones_count += size

print ("Number of 1s in Last",k,"bits are ",ones_count)
```

B.2 Input and Output:

```
Enter Elements : 1 0 0 1 1 1 1 1 0 0 1 0
Bucket Indexes Are :  [(0, 5, 4), (6, 7, 2)]
Buckets are   1 0 0 1 1 1   1 1 0 0 1 0
Enter window size : 10
Number of 1s in Last 10 bits are  4
```

B.3 Observation and learning:

DGIM is an efficient algorithm in processing large streams. When it's infeasible to store the flowing binary stream, DGIM can estimate the number of 1-bits in the window.

B.4 Conclusion:

Successfully implemented DGIM algorithm.

B.5 Questions of Curiosity:

1. Employ the DGIM algorithm. Shown below is a data stream with N=24 and the current bucket config. A new element enters the window at the right. Thus, the oldest bit of the window is the left-most bit shown.

101011000101110110010110

A. What is the largest bucket size for N=24?
B. Show one way of how the above initial stream will be divided into buckets.
C. What is the estimate of the no. of 1's in the latest k=14 bits of this window?

Ans:

A. 4
B.

$$\underset{2^2=\mathbf{4}}{\underline{101011}} \quad 000 \quad \underset{2^2}{\underline{10111}} \quad 0 \quad \underset{2^1}{\underline{11}} \quad 00 \quad \underset{2^1}{\underline{101}} \quad \underset{2^0}{\underline{1}} \quad 0$$

C. 8