Counting oneness in a window:

For this purpose, we are having an algorithm called DGIM algorithm (Datar-Gionis-Indyk-Motwani Algorithm). It is designed to find the number 1's in a data set. It allows to estimate the number of 1's in the window with error of no more than 50%.

DGIM algorithm mainly **consists of two components**

(a) Timestamp: each bit that arrives has a timestamp, for the position at which it arrives. if the first bit has a timestamp 1, the second bit has a timestamp 2 and so on.. the positions are recognized with the window size N (the window sizes are usually taken as a multiple of 2).

(b) Buckets:

(c) The earlier considered

window is divided into buckets consisting of 1's and 0's.

How to form Buckets: Following rules are followed

1.The right side of the bucket should always start with 1. (if it starts with a 0,it is to be neglected)

E.g. · 1001011 → a bucket of size 4 ,having four 1's and starting with 1 on it's right end.

2. Every bucket should have at least one 1, else no bucket can be formed.

3. All buckets should be in powers of 2.

4. The buckets cannot decrease in

size as we move to the left.

(move in increasing order towards left)

Example: Suppose window size N = 24

50 55 59 63 66 69 73

Data → . . 1 0 1 0 1 1 0 0 0 1 0 1 1 1 0 1 1 0 0 1 0 1 1 0 . .

Dividing this window into bucket keeping rules in our mind

Time Stamp 55 63 66 71 72

. . . . 1 0 1 0 1 1 0 0 0 1 0 1 1 1 0 1 1 0 0 1 0 1 1 0 . . .

2^2 2^2 2^1 2^1 2^0

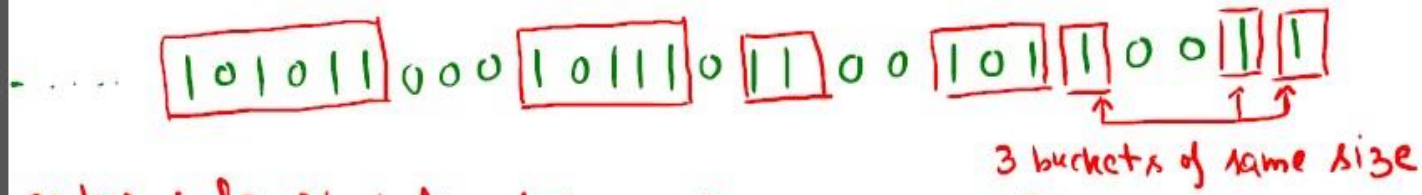
Now suppose new bit 0 is arriving from the right. There is no change

. . . 1 0 1 0 1 1 0 0 0 1 0 1 1 1 0 1 1 0 0 1 0 1 1 0 0 ← . . .

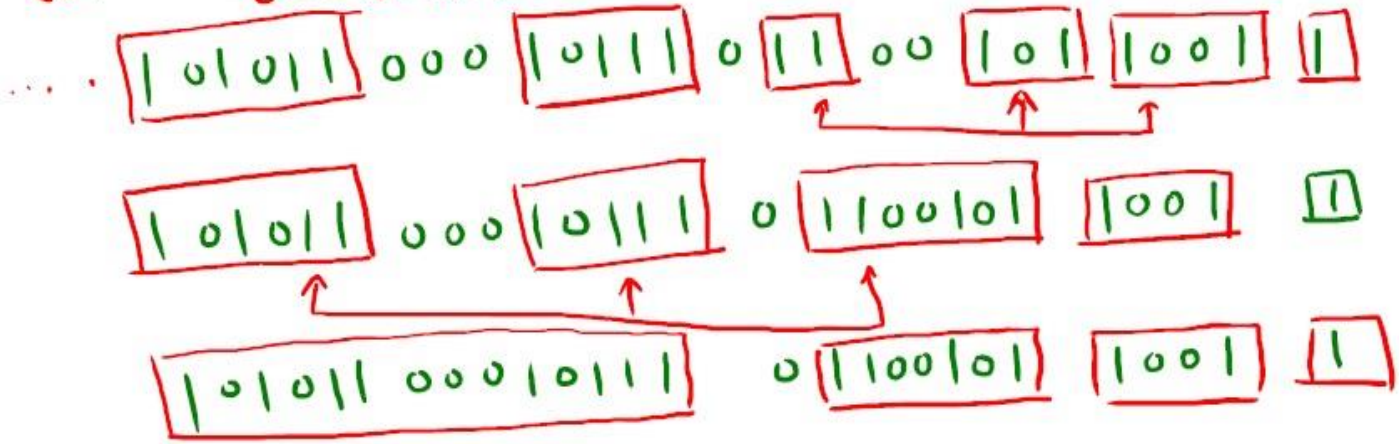
Now suppose new bit is 1

. . . 1 0 1 0 1 1 0 0 0 1 0 1 1 1 0 1 1 0 0 1 0 1 1 0 0 1 ← . . .

Now suppose again 1 bit arrives



as per rule 3 buckets of same size are not allowed therefore merge left most two buckets



- Create a new bucket with the current timestamp and size 1.
- · If there was only one bucket of size 1, then nothing more needs to be done. However, if there are now three buckets of size 1(buckets with timestamp 100,102, 103 in the second step in the picture) We fix the problem by combining the leftmost(earliest) two buckets of size 1. (purple box)
- To combine any two adjacent buckets of the same size, replace them by one bucket of twice the size. The timestamp of the new bucket is the timestamp of the rightmost of the two buckets.
- Now, sometimes combining two buckets of size 1 may create a third bucket of size 2. If so, we combine

  the leftmost two buckets of size 2

  into a bucket of size 4. This process

  may ripple through the bucket sizes.

- How long can you continue doing this…
- You can continue if current timestamp- leftmost bucket timestamp of window < N (=24 here)

E.g. 103–87=16 < 24 so continue, if it greater or equal to then I stop.

Question:

How many 1's are there in the last 20 bits?

Answer:

Counting the sizes of the buckets in the last

20 bits, we say, there are 11 ones.

References:

https://bit.ly/3aIU0qJ

Thank You