

DGIM algorithm

Datar-Gionis-Indynk-Motwani Algorithm
Counting 1's in a stream

Amazon example:

- For every product **X** we keep 0/1 stream of whether that product was sold in the **n**-th transaction
- We want answer queries, how many times have we

Sliding Window: 1 Stream

- Sliding window on a single stream:

N = 6

q w e r t y u i o p a s d f g h j k l z x c v b n m

q w e r t y u i o p a s d f g h j k l z x c v b n m

q w e r t y u i o p a s d f g h j k l z x c v b n m

q w e r t y u i o p a s d f g h j k l z x c v b n m

← Past Future →

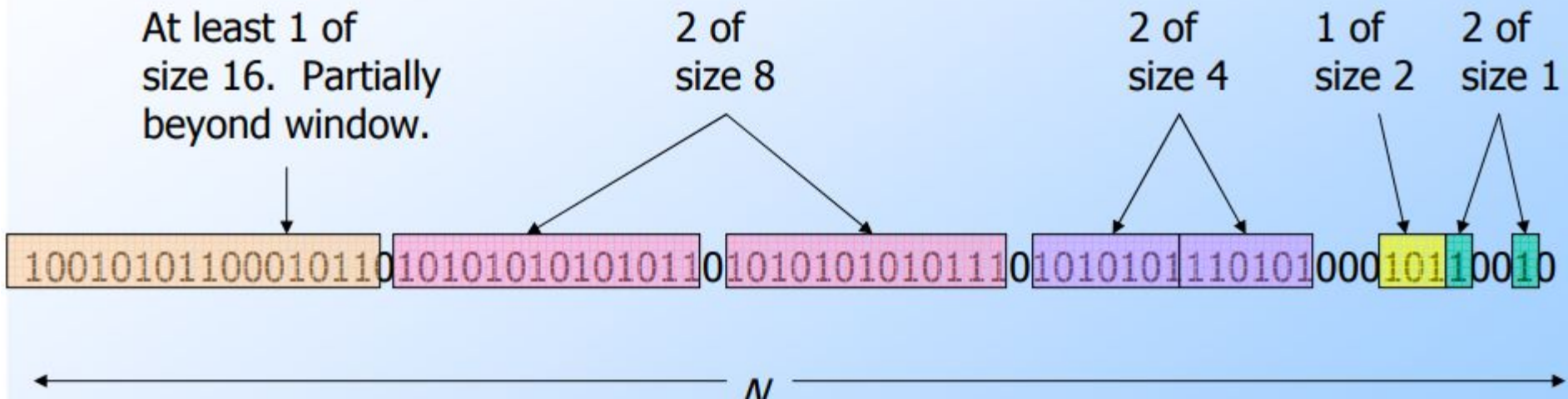
DGIM Algorithm

Representing a Stream by Buckets

- ◆ Either one or two buckets with the same power-of-2 number of 1's.
- ◆ Buckets do not overlap in timestamps.
- ◆ Buckets are sorted by *size* (# of 1's).
 - ◆ Earlier buckets are not smaller than later buckets.
- ◆ Buckets disappear when their end-time is $> N$ time units in the past.

There can be 2 buckets of size 2 or 1 buckets of size 2, as in this case.

Example



Example 2

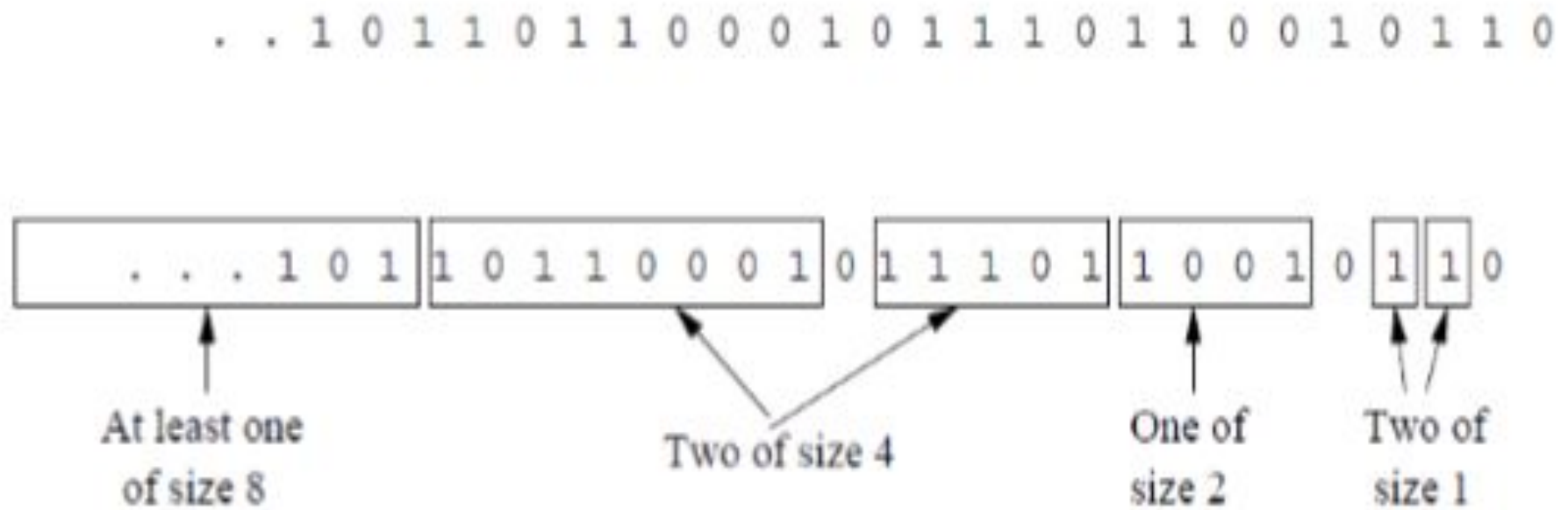


Figure: A bit-stream divided into buckets following the DGIM rules

Updating Buckets --- (1)

- ◆ When a new bit comes in, drop the last (oldest) bucket if its end-time is prior to N time units before the current time.
- ◆ If the current bit is 0, no other changes are needed.

Updating Buckets when 0 comes in

Example

100101011000101101010101010101101010101010111010101011101010100010110010010

Updating Buckets when 0 comes in

Example

100101011000101101010101010101101010101010111010101011101010111010100010110010

Updating Buckets when 1 comes in

Current state of the stream:

100101011000101101010101010101101010101010111010101011101010111010100010110010

Bit of value 1 arrives

001010110001011010101010101011010101010101110101010111010101000101100101

Two orange buckets get merged into a yellow bucket

0010101100010110101010101010110101010101011101010101110101000101100101

Next bit 1 arrives, new orange bucket is created, then 0 comes, then 1:

0101100010110101010101010101101010101011101010101110101000101100101101

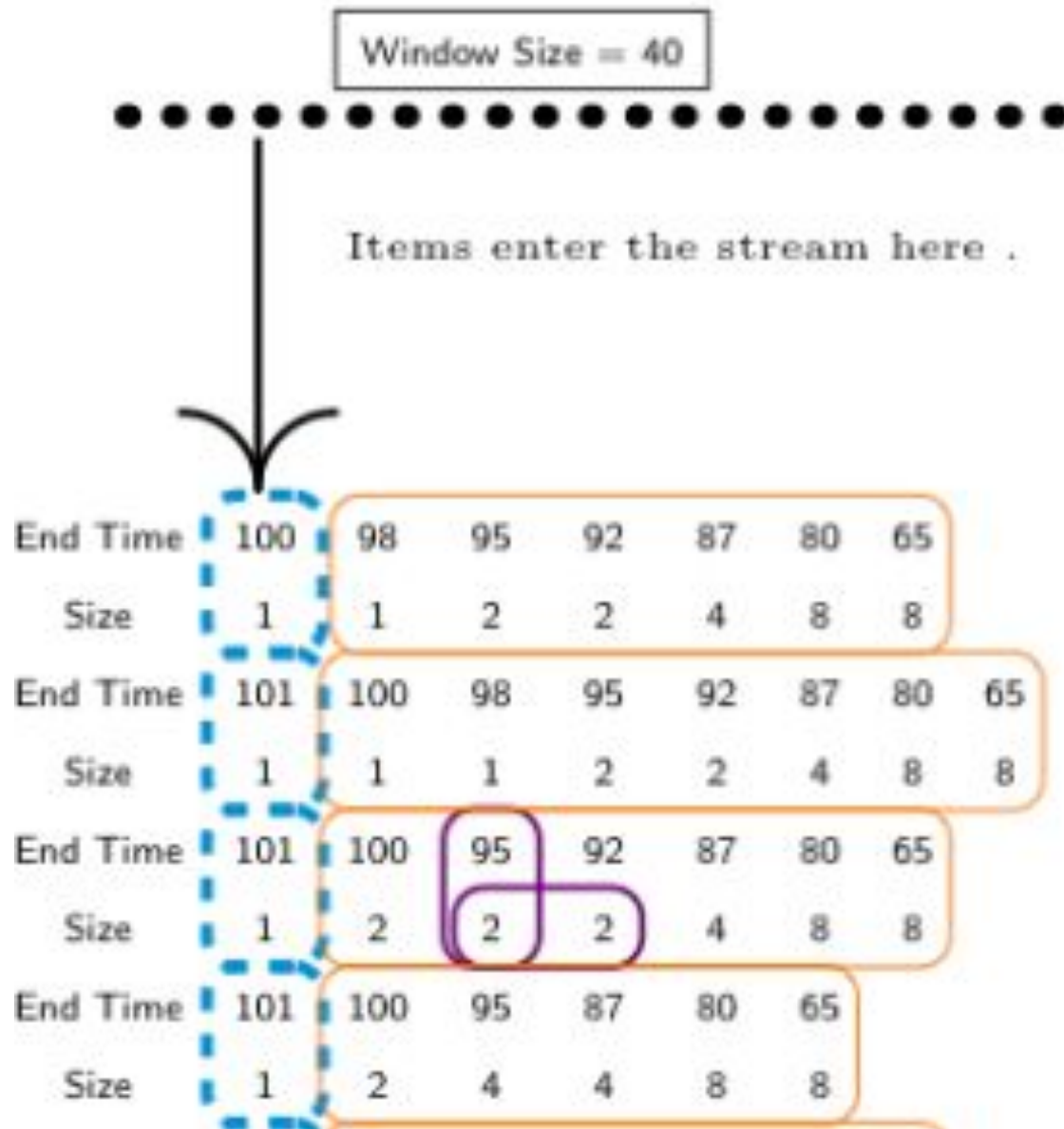
Buckets get merged...

0101100010110101010101010101101010101011101010101110101000101100101101

State of the buckets after merging

0101100010110101010101010101101010101011101010101110101000101100101101

Sliding window of length 40. The current timestamp is 100. Suppose that at times 101 through 105, 1's appear in the stream. Compute the set of buckets that would exist in the system at time 105.



End Time	102	101	100	95	87	80	65		
Size	1	1	2	4	4	8	8		
End Time	103	102	101	100	95	87	80	65	
Size	1	1	1	2	4	4	8	8	
End Time	103	102	100	95	87	80	65		
Size	1	2	2	4	4	8	8		
End Time	104	103	102	100	95	87	80	65	
Size	1	1	2	2	4	4	8	8	
End Time	105	104	103	102	100	95	87	80	65
Size	1	1	1	2	2	4	4	8	8
End Time	105	104	102	100	95	87	80		
Size	1	2	2	2	4	4	8		
End Time	105	104	102	95	87	80			
Size	1	2	4	4	4	8			
End Time	105	104	102	95	80				
Size	1	2	4	8	8				

The difference between the latest timestamp(105) and the oldest(65) equals the windows size(40). So the oldest bucket is dropped .

