5112

10-26

Streaming Algorithms

Streaming

Input : a stream of items

A B A C D F A B A G B C $\longrightarrow$

Hard because we see one item at a time, and can't store them all.

Output: Something about the stream

e.g. All items that occur $\geq 1\%$ of the time.

e.g. # of distinct items.

Majority

Input : a stream

Output : the item that occurs a majority of the stream if it exists

anything otherwise.

Boyer - Moore

Store 1 candidate item a and a counter c.

Case $a \neq \emptyset$ If the ith item $b_i = a$ then $c \leftarrow c+1$
Otherwise $c \leftarrow c-1$.

Otherwise $a \leftarrow b_i$, $c \leftarrow 1$.

Correctness of Boyer-Moore

Assume there is a majority item d.

we're done b/c output d

Let's look at the first time the counter goes back to 0.

other case

If first candidate is d, then maybe this never happens.

Otherwise, eventually counter goes to 0.

So in that case, we had some candidate a.

Saw a k times, and saw "not a" k times.

$\Rightarrow$ saw $d \leq k$ times.

$\Rightarrow$ d is still the majority element in the rest of the stream

$\Rightarrow$ we eventually get to the other case and output d (by induction)

Boyer - Moore Example

A B A C D F A B A G B C

time	a	c			
0	∅	0	6	∅	0
1	A	1			
2	∅	0			
3	A	1			
4	∅	0			
5	D	1			

Frequent Items

Input: A stream of length N.

Output: All items that appear $> \varepsilon N$ times

The version we're going to solve is to return a superset of size $1/\varepsilon$.

Misra-Gries

Set $k = 1/\varepsilon$.

Have k candidates $a_1, \ldots, a_k$ and k counters $c_1, \ldots, c_k$

To process the next item, b:

Case: some $a_i = b$: $c_i \leftarrow c_i + 1$

Otherwise: if some $a_i = \emptyset$: Set $a_i = b$, $c_i = 1$

else: Decrement ALL c_i

A B A C B G B B A A H A B

a_1: A $\quad c_1$: $\cancel{1}\ \cancel{2}\ \cancel{1}\ \cancel{2}\ 3$

a_2: B $\quad c_2$: $\cancel{1}\ 2\ \cancel{1}\ \cancel{2}\ 3$

a_3: H $\quad c_3$: $\cancel{1}\ \cancel{0}\ 1$

Correctness of Misra-Gries

How many decrements can we perform?
Answer at most N/k.

Why? The sum $\sum_{i=1}^{k} c_i \geq 0$.

Each increment increments the sum.
Each decrement subtracts k from the sum.

$\Rightarrow$ Must have $\geq k$ increments for every decrement.

We need to return items that occur at least N/k times

So let d be an item that occurs $> N/k$ times.

Let's set $c(d)$ to be the count for d if positive and 0 otherwise.

$$
\begin{aligned}
c(d) = {} & \underbrace{\#\text{times we see in the stream}}_{> N/k} \\
& \underbrace{- \#\text{times we decrement } d - \#\text{times } d \text{ decrements other things.}}_{< N/k}
\end{aligned}
$$

$\Rightarrow c(d) > 0$ at the end of the stream.

Count Min Sketch

To process b, for each row hash b to a column and increment that counter

	1	2	3	4	5	
1	0	0	0	~~0~~ 1	0	h_1
2	0	~~0~~ 1	0	0	0	h_2
3	0	0		1		h_3
4			1			
5					1	
6	1					

Output the count for d: output the minimum counter for d.