

# Finding Frequent Elements in a Stream (Heavy Hitters)

Jeff M. Phillips

Let  $\mathcal{A}$  be a stream of  $n$  elements  $\langle a_1, \dots, a_m \rangle$  where each  $a_i$  has an integer value in the range  $[1, m]$ . We are allowed one pass on the elements, meaning that we can view each stream element once, in order. We are interested in answering queries on  $\mathcal{A}$  that use space sublinear in  $n$  and  $m$ . Note, many problems are trivial if they are allowed  $O(n)$  or  $O(m)$  space.

We first solve the problem of detecting if any value  $j \in [0, m]$  occurs at least  $\frac{1}{2}n$  times using only a constant amount of space. A  *$\theta$ -heavy hitter* is a value in  $[1, m]$  that occurs more than  $\theta n$  times. The first simple algorithm can solve for all  $\frac{1}{2}$  heavy hitters, of which there are at most 2.

Next we will extend this technique to finding all  $\theta$ -heavy hitters using  $O(1/\theta)$  space and amortized constant time per stream item. Finally, we show how to perform the same algorithm with worst case constant time per stream element.