Advanced Algorithms	11/09, 2017

Exercise 08

Lecturer: Mohsen Ghaffari — *Teaching Assistant: Sebastian Brandt*

1 Majority

In the class, we discussed the deterministic algorithm of Boyer and Moore which has space complexity $O(\log n + \log m)$ and outputs an element such that if the stream has a majority, the output is equal to this majority. Prove that any deterministic algorithm that is able to decide whether a stream has a majority element or not needs space $\Omega(n)$ bits.

2 Frequent Elements

Extend the majority algorithm of Boyer and Moore, which we discussed in the class, to an algorithm space complexity of $O(k(\log n + \log m))$ bits that outputs k elements that include those that appear in more than $1/k$ fraction of the stream.

3 Morris's Approximate Counting Algorithm

In Morris's approximate counting algorithm, which we discussed in the class, prove that

$$\mathbb{E}[2^{2X_m}] = \frac{3}{2}m^2 + \frac{3}{2}m + 1.$$

4 Pairwise Independent Hashing

Prove that given a prime number p, the random hash function $h_{a,b}(x) : \mathbb{F}_p \to \mathbb{F}_p$ defined as $f(x) = ax + b \ mod\, p$, where a and b are random numbers in $\{0, 1, \ldots, p-1\}$ is a pairwise independent hash function, that is, for every $x, y \in \{0, 2, \ldots, p-1\}$ where $x \neq y$, and every $i, j \in \{0, 1, \ldots, p-1\}$, we have $Pr[f(x) = i \ and \ f(y) = j] = 1/p^2$.

5 Distinct Elements

Consider the algorithm that we saw in the class on November 5, for $(1+\varepsilon)$ approximation of the number of distinct elements. We saw that the probability of undershooting by a $(1-\varepsilon)$ factor is at most some constant $c < 1/2$. Prove that the probability of overshooting by a $(1+\varepsilon)$ factor is also upper bounded by $c < 1/2$.