Exercise Set IX, Advanced Algorithms 2024

These exercises are for your own benefit. Feel free to collaborate and share your answers with other students. Solve as many problems as you can and ask for help if you get stuck for too long. Problems marked * are more difficult but also more fun :).

These problems are taken from various sources at EPFL and on the Internet, too numerous to cite individually.

1 Professor Ueli von Gruyères has worked intensely throughout his career to get a good estimator of the yearly consumption of cheese in Switzerland. Recently, he had a true breakthrough. He was able to design an incredibly efficient randomized algorithm $\mathcal{A}$ that outputs a random value X satisfying

$$\mathbb{E}[X] = c \qquad \text{and} \qquad \mathrm{Var}[X] = c^2\,,$$

where c is the (unknown) yearly consumption of cheese in Switzerland. In other words, $\mathcal{A}$ is an unbiased estimator of c with variance c^2.

Use Ueli von Gruyères' algorithm $\mathcal{A}$ to design an algorithm that outputs a random value Y with the following guarantee:

$$\Pr[|Y - c| \geq \epsilon c] \leq \delta \qquad \text{where } \epsilon > 0 \text{ and } \delta > 0 \text{ are small constants.} \tag{1}$$

Your algorithm should increase the resource requirements (its running time and space usage) by at most a factor $O(1/\epsilon^2 \cdot \log(1/\delta))$ compared to the requirements of $\mathcal{A}$.

(In this problem you are asked to (i) design the algorithm using $\mathcal{A}$, (ii) show that it satisfies the guarantee (1), and (iii) analyze how much the resource requirements increase compared to that of simply running $\mathcal{A}$. Recall that you are allowed to refer to material covered in the course.)

2 Suppose that you are given an insertion only stream of items. For every $k \geq 1$, give an algorithm that at each point in the stream maintains k uniformly random elements from the prefix of the stream sampled without replacement. Your algorithm must use $O(k \log n)$ space.

3 Consider a data stream $\sigma = (a_1, \ldots, a_m)$, with $a_j \in [n]$ for every $j = 1, \ldots, m$, where we let $[n] := \{1, 2, \ldots, n\}$ to simplify notation. For $i \in [n]$ let f_i denote the number of times element i appeared in the stream σ.

We say that a stream σ is *approximately sparse* if there exists $i^* \in [n]$ such that $f_{i^*} = \lceil n^{1/4} \rceil$ and for all $i \in [n] \setminus \{i^*\}$ one has $f_i \leq 10$. We call i^* the *dominant* element of σ. Give a single pass streaming algorithm that finds the dominant element i^* in the input stream as long as the stream is approximately sparse. Your algorithm should succeed with probability at least $9/10$ and use $O(n^{1/2} \log^2 n)$ bits of space. You may assume knowledge of n (and that n is larger than an absolute constant).

4 Alice, Bob and Charlie. Suppose that Alice and Bob have two documents d_A and d_B respectively, and Charlie wants to learn about the difference between them. We represent each document by its word frequency vector as follows. We assume that words in d_A and d_B come from some dictionary of size n, and let $x \in \mathbb{R}^n$ be a vector such that for every word $i \in [n]$[1] the entry x_i equals the number of times the i-th word in the dictionary occurs in d_A. Similarly, let $y \in \mathbb{R}^n$ be a vector such that for every word $i \in [n]$ the entry y_i denotes the number of times the i-th word in the dictionary occurs in d_B. We assume that the number of words in each document is bounded by a polynomial in n.

Suppose that there exists $i^* \in [n]$ such that for all $i \in [n] \setminus \{i^*\}$ one has $|x_i - y_i| \leq 2$, and for i^* one has $|x_{i^*} - y_{i^*}| \geq n^{1/2}$. Show that Alice and Bob can each send a $O(\log^2 n)$-bit message to Charlie, from which Charlie can recover the identity of the special word i^*.

Your solution must succeed with probability at least 9/10. You may assume that Alice, Bob and Charlie have a source of shared random bits.

[1] We let $[n] := \{1, 2, \ldots, n\}$.