Advanced Algorithms 2024 28/10, 2024

Exercise 06

Lecturer: Maximilian Probst *Teaching Assistant: Patryk Morawski*

1 Reducing the variance

You are given a randomized algorithm $\mathcal{A}$ that returns a (random) unbiased estimate X for a value x, i.e., $\mathbb{E}[X] = x$. Moreover, you know that $\text{Var}[X] \leq 10^6 x^2$. Design an algorithm that returns a 1.001-approximation for x with probability at least 0.99.

2 Morris's Approximate Counting Algorithm

Recall that in the analysis of the Morris's Aprroximate Counting Algorithm we used a bound on the variance that we have not proved in this lecture. In this exercise we ask you to close this gap. We let Y_m be defined the same as in the lecture notes.

1. Prove that $\mathbb{E}[2^{2Y_m}] = \frac{3}{2}m^2 + \frac{3}{2}m + 1$.
2. Conclude that $Var[2^{Y_m} - 1] \leq \frac{m^2}{2}$ for $m \geq 2$.

3 Bounding the Probability of a Collision in FM+

In the FM+ algorithm, a second hash function g from the family $\mathcal{G}_{2,n,b}$ is used to reduce the memory space required to store the bag $\mathcal{B}$ during the algorithm's execution. The parameter b is set to $b = C \cdot \epsilon^{-4} \cdot \log^2(n)$, where C is a sufficiently large constant that we will determine in this exercise.

We say that an element $i \in [n]$ appeared in the bag $\mathcal{B}$ at some point in the algorithm if, when processing some $a_i = i$, we executed the line $\mathcal{B} \leftarrow \mathcal{B} \cup \{(g(a_i), ZEROS(h(a_i)))\}$. The above approach works as long as there are no collisions between the elements $i \in [n]$ that appeared in $\mathcal{B}$, i.e., if, i.e., if $i \neq j$ have both appeared in $\mathcal{B}$, then it must hold that $g(i) \neq g(j)$.

In this exercise, we will show that the probability that there exist distinct i, j that appeared in $\mathcal{B}$ with $g(i) = g(j)$ is at most $\frac{1}{6}$.

1. Let $S \subseteq [n]$ be of size at most $C' \cdot \epsilon^{-2}(\log n + 2)$. Show that there exists a constant C such that if g is drawn from $\mathcal{G}_{2,n,b}$ with $b = C\epsilon^{-4}\log^2 n$ then the probability that there exist disjoint $i, j \in S$ such that $g(i) = g(j)$ is at most $\frac{1}{6}$.
2. Let $S \subseteq [n]$ be the set of $i \in [n]$ that appeared in $\mathcal{B}$ during this execution and assume that $g(i) \neq g(j)$ for any distinct $i, j \in S$. Show that $|S| \leq C' \cdot \epsilon^{-2}(\log n + 2)$.
3. Using the two observations, conclude by proving that the probability that there exist distinct $i, j \in [n]$ that appeared in $\mathcal{B}$ with $g(i) = g(j)$ is at most $\frac{1}{6}$.

4 Majority Element

Consider the following setting. The elements of the stream $a_1, \ldots, a_m \in [n]$ arrive one-by-one. We want to find an element $j \in [n]$ appearing strictly more that $\frac{m}{2}$ times in the stream. If such an element doesn't exist, we return "DOESN'T EXIST".

1. Assume that such an element j exists. Design a streaming algorithm that finds j. It is only allowed to use $\mathcal{O}(\log n + \log m)$ space.

   Hint (Definition of the algorithm):

   Consider the following algorithm: We keep track of a current guess for j and a counter. Every time a new element arrives, we check whether it's the same as our guess. If it is, we increment the counter by 1. If it isn't, we decrease the counter by 1. If the counter dropped to 0, we replace the current guess with the new element and set the counter to 1 again.

   Prove the correctness of this algorithm.

2. Show that every algorithm deciding whether such an element j exists or not requires at least $\Omega(n)$ space.