CS 170 HW 14

Due **2020-05-06, at 10:00 pm**

1 Study Group

List the names and SIDs of the members in your study group. If you have no collaborators, you must explicitly write none.

2 Opting for releasing your solutions

We are considering releasing a subset of homework submissions written by students for students to see what a full score submission looks like. If your homework solutions are well written, we may consider releasing your solution. If you wish that your solutions not be released, please respond to this question with a "No, do not release any submission to any problems". Otherwise, say "Yes, you may release any of my submissions to any problems".

3 Communicating across a galaxy

Alice is given an *arbitrary* string $x \in \{0,1\}^n$ and Bob is given an *arbitrary* string $y \in \{0,1\}^n$. Their goal is output yes if $x = y$ and no otherwise (with "nontrivial" probability). They wish to come up with a communication protocol with as low communication complexity as possible. In this question we sketch the key idea of 3 algorithms to solve this problem; your job is to:

1. formalize each algorithm sketch,
2. analyze and upper bound the asymptotic communication complexity of your formalization of the algorithm,
3. lower bound the correctness probability of each algorithm.

The algorithms are as follows:

(a) **Prime number based idea.** Let n_x and n_y be the numbers that strings x and y represent in binary respectively.

- Alice chooses a uniformly random prime $\boldsymbol{p}$ in the interval $[1, n^2]$.
- Alice computes $f_x \coloneqq n_x \bmod \boldsymbol{p}$.
- Alice sends f_x and $\boldsymbol{p}$ to Bob.
- Bob computes $f_y \coloneqq n_y \bmod \boldsymbol{p}$.
- If $f_x = f_y$, Bob sends 1 to Alice; otherwise Bob sends 0 to Alice.

You may use that for any N the number of primes less than N is $\Theta\left(\frac{N}{\ln N}\right)$ without proof.

(b) **Polynomials based idea.**

- Alice chooses *any* prime p between n^2 and $2n^2$.
- Alice chooses a uniformly random number $\boldsymbol{r} \sim [0, p]$.
- Alice computes $f_x \coloneqq \sum_{i=0}^{n-1} x_i \boldsymbol{r}^i \bmod p$.
- Alice sends p, f_x and $\boldsymbol{r}$ to Bob.
- Bob computes $f_y \coloneqq \sum_{i=0}^{n-1} y_i \boldsymbol{r}^i \bmod p$.
- If $f_x = f_y$, Bob sends 1 to Alice; otherwise Bob sends 0 to Alice.

You may use the fact that any nonzero polynomial of degree d evaluates to 0 mod p *for at most d inputs in* $[0, p-1]$.

(c) **Anticoncentration based idea.** In this part assume Alice and Bob both have access to the *same set* of r independent strings of length n of uniform and independent ± 1 entries $\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \ldots, \boldsymbol{z}^{(r)}$.

- Alice computes $f_x^{(i)} \coloneqq \sum_{j=1}^{n} x_j \boldsymbol{z}_j^{(i)}$ for $i = 1, \ldots, r$.
- Alice sends $f_x^{(1)}, \ldots, f_x^{(r)}$ to Bob.
- Bob computes $f_y^{(i)} \coloneqq \sum_{j=1}^{n} y_j \boldsymbol{z}_j^{(i)}$ for $i = 1, \ldots, r$.
- If $f_x^{(i)} = f_y^{(i)}$ for $i = 1, \ldots, r$ Bob sends 1 to Alice; otherwise Bob sends 0 to Alice.

Please provide your lower bound on the correctness probability as a function of r.

4 Era of Ravens

(a) Design an algorithm that takes in a stream $z_1, \ldots, z_M$ of M integers in $[n]$ and at any time t can output a uniformly random element in $z_1, \ldots, z_t$. Your algorithm may use at most polynomial in $\log n$ and $\log M$ space. Prove the correctness and analyze the space complexity of your algorithm. Your algorithm may only take a single pass of the stream. *Hint:* $\frac{1}{t} = 1 \cdot \frac{1}{2} \cdot \frac{2}{3} \cdot \frac{3}{4} \cdots \frac{t-1}{t}$.

(b) For a stream $S = z_1, \ldots, z_{2n}$ of $2n$ integers in $[n]$, we call $j \in [n]$ a *duplicative element* if it occurs more than once. Prove that S must contain a duplicative element, and design an algorithm that takes in S as input and with probability at least $1 - \frac{1}{n}$ outputs a duplicative element. Your algorithm may use at most polynomial in $\log n$ space. Prove the correctness and analyze the space complexity of your algorithm. Your algorithm may only take a single pass of the stream.

5 Evasions

In this problem for a hash function $h \in H$, we refer to $n_h(j)$ as the number of elements i such that $h(i) = j$.

(a) Let H be the set of *all* functions from $[n] \to [n]$. Suppose we choose a uniformly random $\boldsymbol{h}$ from H, show that except with probability $o_n(1)$,

$$\max_j n_{\boldsymbol{h}}(j) \leq O\left(\frac{\log n}{\log \log n}\right).$$

Hint: You may use the fact that $\binom{n}{t} \leq \left(\frac{en}{t}\right)^t$ *without proof.*

(b) Let H be a 2-universal hash family. Suppose we choose a uniformly random $\boldsymbol{h}$ from H, show that except with probability $\frac{1}{t^2}$,

$$\max_j n_{\boldsymbol{h}}(j) \leq Ct\sqrt{n}$$

for some absolute constant $C > 0$.