COMPSCI 614: Problem Set 2

Due: 3/6 by 11:59pm in Gradescope.

Instructions:

- You are allowed to, and highly encouraged to, work on this problem set in a group of up to three members.
- Each group should **submit a single solution set**: one member should upload a pdf to Gradescope, marking the other members as part of their group in Gradescope.
- You may talk to members of other groups at a high level about the problems but **not work through the solutions in detail together**.
- You must show your work/derive any answers as part of the solutions to receive full credit.

Hint: The following two inequalities may be helpful throughout the course: for any $x > 0$, $(1+x)^{1/x} \leq e$ and $(1-x)^{1/x} \leq 1/e$.

1. Iterated Balls into Bins (7 points)

1. (1 point) Consider throwing m balls into n bins where we may not have m equal to n. Let b be the number of balls that land in their own bins. What is $\mathbb{E}[b]$?
2. (2 points) Consider the setting of part (1). Assume that $m \leq n$. Prove that $\mathbb{E}[m-b] \leq m^2/n$.
3. (2 points) Consider the following iterated balls-into-bins process: in round 1, throw n balls into n bins and remove any that land in their own bin. In round 2, throw the remaining balls again into the bins and again remove any that land in their own bin. Repeat until all balls are removed. Prove that, if b (from parts (1)-(2)) is exactly as expected in every round, that this process converges in $O(\log \log n)$ rounds.
4. (2 points) Use the above result to give an $O(\log \log n)$ round randomized routing protocol in the setting of Problem 5 on Pset 1. Again, for simplicity you may assume that in each round, the number of balls that land in their own buckets, b, is exactly equal to its expectation.

2. Improved Bound on Distinct Prime Factors (4 points)

Prove that any integer z with $|z| \leq 2^n$ has at most $O\left(\frac{n}{\log n}\right)$ distinct prime factors. This improves the bound of n that we showed in class by a $O(\log n)$ factor. **Hint:** Use that by the prime number theorem, for any x, there are $O(x/\log x)$ primes $\leq x$.

How does this improved bound affect how large a prime we must chose for Rabin Fingerprinting? Will the improved analysis significantly affect our complexity bounds for problems like equality testing via fingerprinting?

3. Public vs. Private Randomness (12 points)

Consider the *public coin* communication complexity model, where Alice and Bob have access to a shared source of randomness. This means e.g., that when using Rabin Fingerprints to check equality of inputs, they do not need to use any communication to share the random prime number p used in the hash function.

1. (3 points) Consider the following public coin protocol for testing equality of two n-length bit strings a and b held by Alice and Bob. Alice and Bob pick a binary string $r = \{0,1\}^n$ uniformly at random. Alice computes $\langle a, r\rangle \mod 2$ and sends this value to Bob. Bob computes $\langle b, r\rangle \mod 2$, and sends 'YES' to Alice if $\langle a, r\rangle \equiv \langle b, r\rangle \mod 2$. He sends 'NO' otherwise. Prove that this protocol is always correct if $a = b$ and is correct with probability at least 1/2 if $a \neq b$.

2. (1 point) Use the above protocol to give a public coin protocol for equality that uses just $O(\log(1/\delta))$ bits of communication and succeeds with probability at least $1 - \delta$. This improves on the protocol based on Rabin Fingerprinting that we covered in class which uses $O(\log(n/\delta)) = O(\log n + \log(1/\delta))$ bits of communication.

We will now show that there is never more than an additive $O(\log(n/\delta))$ factor gap (i.e., the same gap we see above) between the public and private coin communication models.

3. (4 points) Let $\mathcal{P}(r)$ be a public coin communication protocol where the players generate a public random string r and use it to solve some problem on n-bit input strings a and b with probability at least $1 - \delta$ for some $\delta \leq 1/2$. Let $r_1, \ldots, r_t$ be a set of $t = O(n/\delta^2)$ random strings generated by the protocol. Let I_{a,b,r_i} be an indicator random variable which is equal to 1 if $\mathcal{P}(r_i)$ returns the correct answer on inputs $a, b \in \{0,1\}^n$. Prove that for any fixed a, b, $\frac{1}{t}\sum_{i=1}^{t} I_{a,b,r_i} \geq 1 - 2\delta$ with probability at least $1 - \frac{1}{2^{2n+1}}$. I.e. the protocol is correct for at least a $1 - 2\delta$ fraction of the random strings with extremely high probability.

4. (4 points) Use the above result to give a private coin protocol solving the same problem as $\mathcal{P}$ with success probability at least $1 - 2\delta$ and communication complexity at most $O(\log(n/\delta))$ larger than that of $\mathcal{P}$. **Hint 1:** Argue via a union bound that there exists some set of strings $r_1, \ldots, r_t$ where the bound $\frac{1}{t}\sum_{i=1}^{t} I_{a,b,r_i} \geq 1 - 2\delta$ of part (5) holds for *all possible input strings* a, b. **Hint 2:** Your proof does not need to be 'constructive'. I.e., you do not need to show how to find the strings $r_1, \ldots, r_t$ from Hint 1, and can assume that these strings are 'hard-coded' into Alice and Bob's private coin protocol.

4. More Communication Problems (10 points)

1. (3 points) Consider the following communication problem: Alice and Bob both have n-bit strings a, b. If $a \neq b$, both must output the index of some position on which their inputs *do not match*. If $a = b$, they can output anything they want. Describe a randomized protocol for solving this problem with probability $\geq 2/3$ and give a bound on its communication complexity in terms of bits. You may assume that players have access to a shared source of random bits (see Problem 2).

2. (3 points) Prove a lower bound on the bits of communication needed for any deterministic protocol to solve the above problem. As in class, you may restrict yourself to considering protocols in which the players alternate sending 1-bit at a time.

3. (4 points) Consider the setting discussed in class, where n nodes of a graph know only their neighborhood and would like to send messages to a central server, such that the server can determine if the graph is connected with high probability. Consider the harder problem of directed connectivity: the graph is directed and each node only knows its outgoing edges. There are two nodes, s, t, known to all, and the central server wants to determine if there is a directed path from s to t.

   Show that if the nodes are only allowed one-way communication to the central server, then this problem requires $\Omega(n^2)$ total bits of communication to solve (as opposed to $O(n \cdot \log^c n)$ for undirected connectivity).

   Hint: Try to use an indistinguishability argument like we did for the communication complexity of equality testing in class. It suffices to consider directed acyclic graphs, and you may assume for simplicity that the protocol is deterministic – there is not a significant gap between randomized and deterministic algorithms here.

5. Sketching for Minimum Spanning Tree (6 points)

The ℓ_0 sampling algorithm for graph connectivity described in class in fact does not just determine connectivity, but, if the graph is connected, outputs a spanning tree. Consider the more difficult problem of outputting a *minimum weight spanning tree*. Show how to solve this problem with high probability (i.e., $\geq 1 - 1/n^c$ for some constant c), using just $O(n \log^c n)$ total bits of communication for some constant c. You may use several rounds of interaction with the central server, however any messages sent by the central server back to the nodes must be included in your accounting of the communication complexity. You may assume that edges have positive integer weights bounded by n^c for some constant c. A weight of '0' indicates that the edge is not present in the graph.

Hint: Implement Boruvka's algorithm for minimum spanning tree – instead of picking an arbitrary outgoing edge from each connected component, pick the minimum weight outgoing edge. The challenge is to figure out how each node can identify the minimum weight outgoing edge from their connected component after they start being merged together.