CS369E: Exercises & Problems

Instructions:

(1) Students should complete a selection of exercises — a couple per week on average, say — by the end of the course, according to their interests.

(2) Your solutions are due to the instructor by **Thursday March 19th, 2015.**

(3) Problems marked "(*)" are recommended for being particularly relevant and/or interesting.

(4) There are surely some typos; please notify the instructor of any that you find.

Filling in Lecture Details

Lecture 1

1. (Posted 1/8/15.) Recall the one-pass (streaming) algorithm mentioned in lecture that, for a stream that possesses a majority element (appearing more than $m/2$ times), allegedly terminates with this element. Prove the correctness of this algorithm.

2. (*) (Posted 1/8/15.) In our analysis of the F_2 estimator in lecture, we took the mean of $s = \frac{2}{\epsilon^2 \delta}$ independent estimators to ensure an approximation of $(1 \pm \epsilon)$ with probability at least $1 - \delta$. This problem outlines a smarter "median of means" approach, which improves the approximation of the basic estimator first and the success probability second, rather than both at once.

   (a) Suppose we use only the average of $s_1 = \frac{c_1}{\epsilon^2}$ independent estimators, where c_1 is a sufficiently large constant (independent of all other parameters). Adapt the analysis from lecture to argue the result is a $(1 = \pm\epsilon)$-approximation of F_2 with probability at least $2/3$.

   (b) Now suppose we take $s_2 = c_2 \ln \frac{1}{\delta}$ independent groups of s_1 independent estimators each, where c_2 is a sufficiently large constant (independent of all other parameters). Argue that if $\mu_1, \ldots, \mu_{s_2}$ are the means of the groups, then the median of the μ_i's is a $(1 \pm \epsilon)$-approximation of F_2 with probability at least $1 - \delta$.

   (c) Conclude that there is a randomized streaming algorithm that, with probability at least $1 - \delta$, computes a $(1 \pm \epsilon)$-approximation of F_2 using space $O(\epsilon^{-2}(\log n + \log m) \log \frac{1}{\delta})$.

3. (Posted 1/8/15.) The point of this problem is to outline a simple construction of a small family of 4-wise independent hash functions, as required by the AMS F_2 streaming algorithm. Let $n = |U|$ be the universe size, let $\mathbb{F}$ be a finite field with 2^r elements, with $r \in \mathcal{N}$ and $n < |\mathbb{F}| \leq 2n$. Associate the elements of U with n distinct elements from $\mathbb{F}$ (arbitrarily).

   (a) As a warm up, for a pair $(a, b) \in \mathbb{F}^2$ of coefficients, define

   $$h_{ab}(x) = ax + b$$

   for $x \in U$, where all operations are in the field $\mathbb{F}$. Prove that the family $\mathcal{H} = \{h_{ab} : a, b \in \mathbb{F}^2\}$ is pairwise independent, meaning that for every distinct pair $x, y \in U$, for every image $z, w \in \mathbb{F}$, there is a unique function $h_{ab} \in \mathcal{H}$ with $h_{ab}(x) = z$ and $h_{ab}(y) = w$. That is, $\mathcal{H}$ is *pairwise independent.*

(b) For a 4-tuple $(a, b, c, d) \in \mathbb{F}^4$ of coefficients, define

$$h_{abcd}(x) = ax^3 + bx^2 + cx + d,$$

where all operations take place in the field $\mathbb{F}$. Let $\mathcal{H}$ denote the set of all $|\mathbb{F}|^4$ such functions. Prove that for every 4-tuple (x_1, x_2, x_3, x_4) of distinct elements of U, and every 4-tuple z_1, z_2, z_3, z_4 of images in $\mathbb{F}$, there is a unique function $h_{abcd} \in \mathcal{H}$ with $h_{abcd}(x_i) = z_i$ for $i = 1, 2, 3, 4$.

(c) Define $g_{abcd}(x)$ as +1 if $h_{abcd}(x)$ is an even integer and -1 otherwise. Prove that $\mathcal{G} = \{g_{abcd} : a, b, c, d \in \mathbb{F}\}$ is 4-wise independent, meaning that for all distinct $x_1, x_2, x_3, x_4 \in U$ and all $z_1, z_2, z_3, z_4 \in \{\pm 1\}$,

$$\mathbf{Pr}_{g \in \mathcal{G}}[g(x_i) = z_i \text{ for } i = 1, 2, 3, 4] = \frac{1}{16}.$$

[Note: describing a function g of $\mathcal{G}$ requires only $O(\log n)$ bits, for the four coefficients $a, b, c, d \in \mathbb{F}$. The evaluation of such a hash function can also be carried out with a logarithmic amount of space.]

4. (*) (Posted 1/8/15.) This problem outlines the argument that, for every non-negative integer other than 1, the deterministic approximate computation of F_k requires linear space. That is, randomization is essential to our small-space estimation algorithms for F_0 and F_2.

(a) Let U be a universe of size n. You can assume that n is sufficiently large. Prove that there is a constant $c > 0$ such that there exists a collection of $\mathcal{C} \subseteq 2^U$ of subsets of U with the following properties:

(i) $|\mathcal{C}| \geq 2^{cn}$;

(ii) every subset $S \in \mathcal{C}$ has size $|S| = \frac{n}{4}$

(iii) every pair $S, T \in \mathcal{C}$ of distinct sets has small intersection $|S \cap T| \leq \frac{n}{8}$.

[Hint: use the probabilistic method. That is, choose a bunch of sets S at random and argue, using Chernoff bounds, that the resulting collection $\mathcal{C}$ satisfies (i)–(iii) with positive probability.]

(b) Consider streams of the form (S, T), where S and T are (not necessarily distinct) sets from $\mathcal{C}$, arranged in arbitrary order. Argue that a deterministic streaming algorithm with sublinear space has identical memory contents for two different "first halves" S_1 and S_2.

(c) By considering the cases where $T = S_1$ and $T = S_2$, argue that no deterministic streaming algorithm with sublinear space can always obtain a (1 ± 0.1)-approximation estimate of F_k, where k is a nonnegative integer other than 1.

Lecture 2

1. (Posted 1/26/15.) Suppose that a problem is solved by a public-coin randomized one-way protocol that has two-sided error $\epsilon_1 < \frac{1}{2}$ and uses communication c. Prove for every constant $\epsilon_2 > 0$, the problem is also solved by a public-coin randomized one-way protocol with two-sided error ϵ_2 and communication $O(c)$. What is the dependence on ϵ_1 and ϵ_2 of the constant hidden in the big-Oh notation?

2. (Posted 1/26/15.) Prove the harder direction of Yao's Lemma. That is, suppose that every public-coin randomized one-way protocol R for a problem that has two-sided error at most ϵ has communication cost at least c. Prove that there exists a distribution D over inputs to the problem such that every deterministic protocol P for the problem with error at most ϵ (over the random input) has communication cost at least c.

[Hint: assume and use von Neumann's Minimax Theorem for zero-sum two-player games.]

3. (Posted 1/26/15.) Prove that for every positive integer n and $k \in \{1, 2, \ldots, n\}$,

$$\left(\frac{n}{k}\right)^k \leq \binom{n}{k} \leq \left(\frac{en}{k}\right)^k.$$

For the second inequality, feel free to use *Stirling's approximation*, which states that there is a constant c (namely, $c = \sqrt{2\pi}$) such that, with negligible error,

$$n! \approx c\sqrt{n}\left(\frac{n}{e}\right)^n \tag{1}$$

for all positive integers n.

4. (Posted 1/26/15.) Prove that the probability that $2n$ fair coin flips produce an equal number of "heads" and "tails" is $\approx \frac{c}{\sqrt{n}}$ for a suitable constant c.

   [Hint: use Stirling's approximation (1).]

5. (Posted 1/26/15, corrected 3/6/15.) Fill in this missing details of the Chernoff bound application in the proof of Theorem 7.1. Precisely, consider a coin that comes up "heads" with probability p, where p is either $\frac{1}{2} + \frac{1}{\sqrt{n}}$ (case 1) or $\frac{1}{2} - \frac{1}{\sqrt{n}}$ (case 2). Flip this coin ℓ times to generate ℓ bits. Prove that, provided $\ell = cn$ for a sufficiently large constant c, the probability that the number of heads is at least $\frac{\ell}{2} + \sqrt{\ell}$ (when in case 1) or at most $\frac{\ell}{2} - \sqrt{\ell}$ (when in case 2) is at least $\frac{8}{9}$.

6. (Posted 1/26/15.) (*) Prove that the lower bound of Theorem 7.2 holds also for the problem of approximating F_2.

   [Hint: make minor modifications to the reduction in Section 6.]

7. (Posted 1/26/15.) Extend Theorem 7.2 to show that a $(1 \pm \epsilon)$-approximation of F_0 (with probability at least 2/3) requires $\Omega(\epsilon^{-2})$ space.

   [Hint: just turn the "padding trick" mentioned in the lecture notes into a formal proof.]

Lecture 3

1. (Posted 1/26/15.) Use the Pigeonhole Principle to prove that every deterministic one-way protocol for Equality has communication cost n. Do the same for the Augmented Index problem.

2. (Posted 2/4/15.) Prove Lemma 3.2 from the lecture notes.

   [Hint: use the probabilistic method. Compare to exercise 4(a) of Lecture #1.]

3. (Posted 2/4/15.) Suppose there is an $m \times n$ matrix $\mathbf{A}$ (with linearly independent rows) such that: there exists a constant $c \geq 1$ and a recovery algorithm R that, for every $\mathbf{x} \in \mathbb{R}^n$, computes from $\mathbf{Ax}$ a vector x' such that

   $$\|\mathbf{x}' - \mathbf{x}\|_1 \leq c \cdot \text{res}(\mathbf{x}). \tag{2}$$

   Then, prove that there exists such a matrix $\mathbf{A}$ with orthonormal rows.

4. (*) (Posted 2/4/15.) In our proof of Theorem 3.1, we assumed that the sensing matrix $\mathbf{A}$ has entries that can be described with $O(\log n)$ bits each. Extend the reduction to handle the case of sensing matrices $\mathbf{A}$ with arbitrary entries.

   [Hints: building on the previous exercise, show that $\mathbf{A}$ can be converted (by Alice and Bob, before the protocol starts) into a matrix $\mathbf{A}'$ with entries that are polynomially bounded integers such that given $\mathbf{A}'\mathbf{y}$ Bob can find (by brute-force search) a very small vector $\mathbf{s}$ such that $\mathbf{A}(\mathbf{y} + \mathbf{s}) = \mathbf{A}'\mathbf{y}$. That is, it's as if Alice used the true sensing matrix $\mathbf{A}$ but accidentally used a slightly perturbed version of the intended vector $\mathbf{y}$. Show that, because $\mathbf{s}$ is small, this is good enough for Bob to run the recovery algorithm R (for $\mathbf{A}$) and deduce Alice's input to the Index function.]

Lecture 4

1. (Posted 2/4/15.) Prove that a subset $S \subseteq X \times Y$ is a rectangle — that is, of the form $S = A \times B$ — if and only if S is closed under "mix and match:" whenever $(\mathbf{x}_1, \mathbf{y}_1)$ and $(\mathbf{x}_2, \mathbf{y}_2)$ belong to S, so do $(\mathbf{x}_1, \mathbf{y}_2)$ and $(\mathbf{x}_2, \mathbf{y}_1)$.

2. (Posted 2/4/15.) Let X and Y denote the sets of possible inputs to Alice and Bob, respectively. We showed in lecture that every deterministic protocol induces a partition of $X \times Y$ into rectangles, with one rectangle per protocol transcript (equivalently, per leaf of the protocol tree). Show that the converse fails: there are sets X, Y and a partition of $X \times Y$ into rectangles so that no protocol induces the partition.

   [Hint: Try it with $|X| = |Y| = 3$.]

3. (Posted 2/4/15.) Recall the Clique-Independent Set problem from lecture. Prove that the one-way randomized communication complexity of the problem is $\Omega(n)$.

4. (*) (Posted 2/4/15.) In lecture we used Newman's Theorem to give a private-coin randomized protocol for Equality with constant error and communication cost $O(\log n)$. (One-way, even.) This exercise gives a more direct argument that achieves the same communication cost, and with better error to boot.

   Recall that for every $k > 1$ there is at least one prime number between k and $2k$. Alice and Bob agree on such a prime p in $\{n^2 + 1, \ldots, 2n^2 - 1\}$ in advance. (Or Alice can pick it and send it to Bob later, it doesn't matter.) When Alice receives her input $\mathbf{x} \in \{0, 1\}^n$, she interprets it as the univariate polynomial

   $$a(z) = \sum_{i=1}^{n} x_i z^{i-1}.$$

   She then chooses an evaluation point $t \in \{0, 1, 2, \ldots, p-1\}$ uniformly at random and sends t and $a(t) \bmod p$ to Bob. Bob interprets his input $\mathbf{y}$ as the polynomial

   $$b(z) = \sum_{i=1}^{n} y_i z^{i-1}.$$

   and answers "equal" if $b(z) \bmod p = a(z) \bmod p$ and "not equal" otherwise.

   Prove that this one-way private-coin communication protocol has communication cost $O(\log n)$ and 1-sided error at most $1/n$.

   [Hint: assume and use the fact that, for a prime p, a polynomial of degree d has at most d roots over the field $\mathcal{Z}_p$.]

Lecture 5

1. (Posted 3/15/15.) Prove the Birkhoff-von Neumann Theorem. This theorem states that every fractional perfect matching of a bipartite graph — x_{ij}'s in $[0, 1]$ (with $i \in U, v \in V$) satisfying $\sum_{j \in V} x_{ij} = 1$ for every $i \in U$ and $\sum_{i \in U} x_{ij} = 1$ for every $j \in V$ — is the convex combination of (characteristic vectors of) perfect matchings.

   [Hint: viewing the x_{ij}'s as a matrix, argue that the support of the matrix must include that of a permutation matrix. Then proceed by induction on the number of non-zero x_{ij}'s.]

2. (*) (Posted 3/15/15.) We proved one direction of Yannakakis's Lemma — if there exists an extended formulation Q of a polytope P with only r inequalities, then the slack matrix of P has nonnegative rank at most r. (Recall in the slack matrix S, for a row (face) f and a column (vertex) v, S_{fv} is defined as $b - a^T v$, where $a^T x \leq b$ is a supporting hyperplane inducing the face f.) Prove the converse: if the slack matrix S of P has nonnegative rank r, then there is an extended formulation Q that uses only r inequalities.

Lecture 8

1. (Posted 3/11/15.) Theorem 5.4 gives a lower bound for the *worst* ϵ-Nash equilibrium of the game. Explain why the worst-case lower bound holds more generally for every (non-empty) subset of ϵ-Nash equilibria such that: (i) there is at least one member of the subset with description length polynomial in k, the logarithm of the maximum number of actions of a player, and $\frac{1}{\epsilon}$; and (ii) membership in the subset can be verified by the players without any communication.

2. (*) (Posted 3/11/15.) Does the proof of Theorem 5.4 imply that *every* ϵ-Nash equilibrium with sufficiently small description has expected welfare at most a $1/\alpha$ fraction of the maximum possible? Explain.

3. (Posted 3/11/15.) Consider the welfare-maximization problem with all values $v_i(S)$ polynomially bounded integers. Prove that there is an auction with a doubly exponential number of actions per player such that, for every choice $v_1, \ldots, v_k$ of valuations, there exists an exact Nash equilibrium such that: (i) the welfare is optimal; (ii) the description length is polynomial in k and the logarithm of the maximum number of actions of a player; (iii) the equilibrium can be verified privately by the players (cf., Exercise 1 above).

   [Hint: read about the "VCG mechanism," e.g. in Lecture #7 of the instructor's CS364A course.]

4. (*) (Posted 3/11/15.) Consider a two-player game, where each player has s strategies. Assume that all player utilities are between 0 and 1. Prove that there is an ϵ-Nash equilibrium in which all probabilities are multiples of $1/t$ for an integer $t = O(\epsilon^{-2} \log s)$. [In particular, each player randomizes over at most t different strategies.]

   [Hint: fill in the details of the proof outline given in lecture.]

5. (Posted 3/11/15.) Extend the preceding result to k-player games. Can you achieve t polynomial in k, $\log m$, and $\frac{1}{\epsilon}$? How about polynomial in $\log k$, $\log m$, and $\frac{1}{\epsilon}$?

Lecture 9

1. (Posted 3/15/15.) Recall the distinction between non-adaptive and adaptive testers. Prove that the query complexity of non-adaptive testers is at most exponential in that of adaptive testers.

2. (*) (Posted 3/15/15.) Prove that our analysis of the edge tester is tight in the Boolean case: there exists a Boolean function f that is ϵ-far from monotone, and set the probability that a single random edge find a monotonicity violation is only $O(\epsilon/n)$, where n is the number of coordinates.

3. (*) (Posted 3/15/15.) Prove an upper bound of $O(\frac{n}{\epsilon}) \log |R|)$ on the query complexity of testing monotonicity of functions from $\{0,1\}^n$ to a totally ordered set R.

   [Hint: make precise the "divide and conquer" idea mentioned in class.]

4. (Posted 3/15/15.) Recall we concluded the lecture by proving an $\Omega(n)$ query complexity lower bound for ranges of size $\Theta(\sqrt{n})$ (for constant ϵ). Use a padding argument to obtain a lower bound of $\Omega(|R|^2)$ for small ranges R.

5. (*) (Posted 3/17/15.) We proved Theorem 3.1 for functions with domain $\{0,1\}^n$ and range $\{0,1\}^n$. Suppose we try to re-use the same proof for a function with domain $\{0,1\}^n$ and an arbitrary totally ordered range R. Where exactly (if anywhere) does the proof break? Specifically, does fixing the coordinates one-by-one by swapping function values result produce a monotone function after modifying f in at most $2\sum_{i=1}^{n} |A_i|$ entries?

6. (*) (Posted 3/17/15.) We concluded lecture with a general template for deriving query complexity lower bounds for adaptive testers from communication complexity loewr bounds for general communication protocols. Formulate an analogous template and simulation argument for non-adaptive testers and one-way communication protocols.

Further Communication Complexity

1. (*) (Posted 2/4/15.) It is clear that if every protocol tree of a deterministic protocol that computes a function f has at least t leaves, then the deterministic communication complexity of f is at least $\approx \log_2 t$. But what about the converse? Is it possible that a function f has large communication complexity even though it can be computed by a protocol whose tree has a small number of leaves? In principle, this could happen if all such trees are "scraggly," with a large depth (and hence large worst-case communication) despite having few leaves.

   This exercise rules out this possibility. Prove that if a function can be computed by a protocol with a tree that has at most ℓ leaves, then its deterministic communication complexity is $O(\log \ell)$.

   [Hint: use the fact that every binary tree with t nodes has a "median" — a node such that every one of its subtrees contains at most $\frac{2}{3}t$ nodes. Use this to transform an arbitrary protocol tree into a relatively balanced one.]

2. (*) (Posted 2/4/15.) Recall from Lecture #4 the matrix $M(f)$ of a function f. Prove that the deterministic communication complexity of f is at least $\log_2(2\text{rank}(M(f)) - 1)$, where $\text{rank}(\cdot)$ denotes the rank of the matrix over the reals.

   [Hints: To prove a lower bound of $\log_2 \text{rank}(M(f))$, think of the 1-rectangles in a partition of $M(f)$ into monochromatic rectangles as a decomposition of $M(f)$ into rank 1 matrices. To boost the lower bound to $\log_2(2\text{rank}(M(f)) - 1)$ by getting the 0-rectangles involved, apply the same argument to $\mathbf{J} - M(f)$, where $\mathbf{J}$ is the all-ones matrix.]

3. (*) (Posted 2/4/15.) The *log-rank conjecture*, which remains very much open, speculates a converse to the previous exercise, that the deterministic communication complexity of a function f is at most polylogarithmic in $\log_2 \text{rank}(M(f))$.

   Show that, at the very least, the deterministic communication complexity of a function f is at most $\text{rank}(M(f)) + 1$.

   [Hint: if $M(f)$ has rank r, how many distinct rows can it have?]

4. (Posted 3/15/15.) We defined nondeterministic protocols with a third-party prover, and where Alice and Bob don't communicate directly at all. Formulate an alternative notion of communication protocols where Alice and Bob communicate nondeterministically (without any explicit prover), and prove that the two notions are equivalent.

5. (Posted 3/15/15.) Prove that every communication complexity lower bound for nondeteministic protocols applies also to randomized protocols with 1-sided error.

Further Results: Streaming Algorithms and Lower Bounds

1. (Posted 1/8/15.) *Misra-Gries algorithm.* Suppose you are told that a data stream of length m will contain $k-1$ elements that each appear strictly more than m/k times. Give a generalization of the one-pass algorithm for finding a majority element for finding these $k-1$ elements. Your algorithm should only remember $k-1$ elements at any given time, plus a counter for each. Prove that your algorithm is correct.

2. (*) (Posted 1/8/15.) *Morris's algorithm.* As mentioned in Lecture #1, it is trivial to compute F_1 (i.e., to count) using $\approx \log_2 m$ space, where m is the number of objects being counted. But what if we only care about about counting approximately, up to a $(1 \pm \epsilon)$ factor? Here's a way to reduce the space to $O(\epsilon^{-2} \log \log m \log \frac{1}{\delta})$. The basic idea is to count (probabilistically) $\log m$ rather than m itself, and then aggregate many independent estimates (as in our F_2 analysis).

   (a) The basic estimator is the following. Initialize $Z = 0$. When a new object arrives, increment Z with probability 2^{-Z} (else leave it unchanged). At the end, output $X = 2^Z - 1$.
   Prove that the estimator is unbiased, that $\mathbf{E}[X] = m$.
   [Hint: prove by induction on i that, after seeing i objects, $\mathbf{E}\left[2^Z\right] = 2^i - 1$.]

(b) Prove that $\mathbf{E}\left[2^{2Z}\right] = \frac{3}{2}m^2 + \frac{3}{2}m + 1$.
[Hint: again, induction on i.]

(c) Conclude that $\mathbf{Var}[X] = \frac{n(n-1)}{2}$.

(d) Use the average of several independent estimators and Chebyshev's inequality (as in Lecture #1) and the median trick from Exercise 2 of Lecture #1 to prove that a probabilistic $(1 \pm \epsilon)$-approximate counter requires only $O(\epsilon^{-1} \log\log m \log \delta \frac{1}{\delta})$ space (where δ upper bounds the probability of failing to compute a $(1 \pm \epsilon)$-approximation).

(e) Explain how to use such probabilistic approximate counters to improve the space usage of the F_2 estimation algorithm in lecture from $O(\epsilon^{-2}(\log n + \log m) \log \frac{1}{\delta})$ to $O(\epsilon^{-2}(\log n + \log\log m) \log \frac{1}{\delta})$.