PRINCETON UNIVERSITY F'02 COS 597D: A THEORIST'S TOOLKIT

Homework 2

Out: *October 7* Due: *October 21*

You can collaborate with your classmates, but be sure to list your collaborators with your answer. If you get help from a published source (book, paper etc.), cite that. Also, limit your answers to one page or less —you just need to give enough detail to convince me. If you suspect a problem is open, just say so and give reasons for your suspicion.

§1 Let $\mathcal{A}$ be a class of deterministic algorithms for a problem and I be the set of possible inputs. Both sets are finite. Let $\text{cost}(A, x)$ denote the cost of running algorithm A on input x. The *distributional complexity* of the problem is

$$\max_{\mathcal{D}} \min_{A \in \mathcal{A}} \mathbf{E}_{x \in \mathcal{D}}[\text{cost}(A, x)],$$

where $\mathcal{D}$ is a probability distribution on I. The *randomized complexity* is

$$\min_{\mathcal{P}} \max_{x \in I} \mathbf{E}_{A \in \mathcal{P}}[\text{cost}(A, x)],$$

where $\mathcal{P}$ is a probability distribution on $\mathcal{A}$. Prove Yao's lemma, which says that the two complexities are equal.

Does this result hold if the class of algorithms and inputs is infinite?

§2 (*The rent-or-buy problem*) Your job requires you to take long trips by car. You can either buy a car ($ 5000) or rent one whenever you need it ($ 200 each time).

If i is the number of trips you end up making, clearly, it makes sense to buy if $i > 25$. (Let us ignore emotional factors, as well as the fact that a used car is still worth something after a few years.) Furthermore the optimum expenditure as a function of i is $C(i) = \min\{5000, 200i\}$.

However, you do not know i ahead of time: in fact, you don't learn about each trip until the day before, at which time you have to rent or buy. (Such problems are studied in a field called *online algorithms*.)

(a) Show that if you make the decision to buy or rent at each step using a deterministic algorithm then there is a strategy whose cost is at most $2C(i)$ for any i. Show that no deterministic algorithm can do much better for some i. (Alas, life is suboptimal??)

(b) Does your answer change if you can use a randomized strategy, and try to minimize your expected cost? (Hint: Lowerbounds can be proved using Yao's Lemma.)

§3 Prove the Schwartz-Zippel Lemma: If $g(x_1, x_2, \ldots, x_m)$ is any nonzero polynomial of total degree d and $S \subseteq F$ is any subset of field elements, then the fraction of $(a_1, a_2, \ldots, a_m) \in S^m$ for which $g(a_1, a_2, \ldots, a_m) = 0$ is at most $d/|S|$.

§4 (*Sudan's list decoding*) Let $(a_1, b_1), (a_2, b_2), \ldots, (a_n, b_n) \in F^2$ where $F = GF(q)$ and $q \gg n$. We say that a polynomial $p(x)$ *describes* k of these pairs if $p(a_i) = b_i$ for k values of i.

(a) Show that there exists a bivariate polynomial $Q(z, x)$ of degree at most $\lceil \sqrt{n} \rceil + 1$ in z and x such that $Q(b_i, a_i) = 0$ for each $i = 1, \ldots, n$. Show also that there is an efficient (poly(n) time) algorithm to construct such a Q.

(b) Show that if $R(z, x)$ is a bivariate polynomial and $g(x)$ a univariate polynomial then $z - g(x)$ divides $R(z, x)$ iff $R(g(x), x)$ is the 0 polynomial.

(c) Suppose $p(x)$ is a degree d polynomial that describes k of the points. Show that if d is an integer and $k > (d + 1)(\lceil \sqrt{n} \rceil + 1)$ then $z - p(x)$ divides the bivariate polynomial $Q(z, x)$ described in part (a). (Aside: Note that this places an upper-bound on the number of such polynomials. Can you improve this upperbound by other methods?)

(There is a randomized polynomial time algorithm due to Berlekamp that factors a bivariate polynomial. Using this we can efficiently recover all the polynomials p of the type described in (c). This completes the description of Sudan's algorithm for *list decoding*.)