6.080 / 6.089 Great Ideas in Theoretical Computer Science
Spring 2008

6.080/6.089 Problem Set 4

Assigned: Thursday, April 10, 2008 / Due: Thursday, April 24, 2008

1. Let X be a random variable that takes nonnegative integer values. Show that $\mathrm{E}[X] = \sum_{i=1}^{\infty} \Pr[X \geq i]$. [*Hint:* First convince yourself that this is true by trying special cases.]

2. Suppose a ZPP algorithm succeeds with probability p, and outputs "don't know" with probability $1-p$. Calculate the expected number of times we need to run the algorithm, until it succeeds.

3. Let Y be an n-bit string chosen randomly among all strings with an even number of 1's, and let Y_A be the substring of Y consisting only of bits in positions $A \subseteq \{1, \ldots, n\}$.

   (a) Show that if A and B intersect, then Y_A and Y_B are not independent.

   (b) Show that if A and B are disjoint and $A \cup B \neq \{1, \ldots, n\}$, then Y_A and Y_B *are* independent.

   (c) Show that if A and B are disjoint (and nonempty) and $A \cup B = \{1, \ldots, n\}$, then Y_A and Y_B are not independent.

4. Suppose we have n balls and n buckets, and suppose each ball is thrown into one of the buckets completely at random (independently of all the other balls).

   (a) Let p_n be the probability that at least one ball lands in the first bucket. What is $\lim_{n\to\infty} p_n$?

   (b) Let q_n be the probability that every ball lands in a separate bucket. Show that q_n decreases exponentially with n.

   (c) *[Extra credit]* Let m be the maximum number of balls that land in any one bucket. Show that there's a positive constant c such that $m \leq c \log n$ with high probability. [*Hint:* Use the union bound, combined with the following version of the Chernoff bound. Let $X_1, \ldots, X_n$ be any independent, $\{0,1\}$-valued random variables and let $X = X_1 + \cdots + X_n$. Then $\Pr[X > (1+\delta)\,\mathrm{E}[X]] < \left[e^{\delta} / (1+\delta)^{1+\delta}\right]^{\mathrm{E}[X]}$ for all $\delta > 0$.]

5. Show that, if there's a two-sided-error randomized algorithm that solves NP-complete problems in polynomial time, then there's also a one-sided-error randomized algorithm. Or more concisely, if $\mathsf{NP} \subseteq \mathsf{BPP}$ then $\mathsf{NP} \subseteq \mathsf{RP}$, and hence $\mathsf{NP} = \mathsf{RP}$. [*Hint:* Use the equivalence of search and decision problems from Pset3. Amplification and the union bound could also come in handy.]

6. In many cryptographic applications (for example, digital signature schemes), it's important to have a function $f : \{0,1\}^m \to \{0,1\}^n$ for which it's computationally infeasible to find a *collision*: that is, two distinct inputs x and y such that $f(x) = f(y)$. Such an f is called a *collision-resistant hash function.* Here you should think of m as much larger than n.

   (a) Suppose f is chosen uniformly at random, and suppose the only way an algorithm can learn about f is by calling a subroutine that evaluates $f(x)$ on any given input x. Show that, on average, the algorithm will need to call the subroutine $\Omega\left(2^{n/2}\right)$ times before it finds a collision. [*Hint:* Use the union bound.]

   (b) *[Extra credit]* Show that for any such function f, after evaluating f on only $O\left(2^{n/2}\right)$ randomly-chosen values, with high probability we will have found a collision.

7. Show that there is no one-way function where every bit of the output depends on only two bits of the input. [*Hint:* Use the fact that $2SAT$ is in P.]