Randomized Algorithms (Blum/Gupta) **Homework 2**
Date: Jan 26, 2011 **Due:** Feb 9, 2011

Groundrules

- Homeworks will generally consist of *exercises*, easier problems designed to give you practice, and *problems*, that may be harder, trickier, and/or somewhat open-ended. You should do the exercises by yourself, but you may work with a friend on the harder problems if you want. One exception: no fair working with someone who has already figured out (or already knows) the answer. If you work with a friend, then write down who you are working with.
- If you've seen a problem before (sometimes we'll give problems that are "famous"), then say that in your solution (it won't affect your score, we just want to know). Also, if you use any sources other than the textbook, write that down too (it's fine to look up a complicated sum or inequality or whatever, but don't look up an entire solution).

Reading: Listed next to the lectures on the webpage.

Exercises

1. **(Circuits and Coins.)** Prove that **BPP** $\subseteq$ **P/poly**.
2. **(Quick! Select!)** Quickselect is the following simple algorithm for finding the kth smallest element in an unsorted set S (in the book, this is called Find and is described on p.15).

   Quickselect(S, k):

   (a) Pick a pivot element p at random from S.

   (b) By comparing p to each element of S, split S into two pieces: $S_1 = \{x \in S : x < p\}$ and $S_2 = \{x \in S : x > p\}$.

   (c) If $|S_1| = k-1$ then output p.
   If $|S_1| > k-1$ then output Quickselect(S_1, k).
   If $|S_1| < k-1$ then output Quickselect$(S_2, k - |S_1| - 1)$.

   Prove that the expected number of comparisons made by Quickselect on a set S of size n is at most $3.5\,n$. (Half credit for proving it's at most $4n$.) You may assume that initially $k = n/2$ (we are trying to find the median of S), which is the worst case.
3. **(The Markov Success.)** The analysis of the algorithm for Max-3-SAT showed that a random setting of the variables satisfied a 7/8-fraction of the clauses in expectation. Using Markov's Inequality, show that for any $\epsilon > 0$, repeating the randomized algorithm $t = O(1/\epsilon)$ times and taking the best of the t solutions satisfies at least $(7/8) - \epsilon$ fraction of the clauses with probability at least $1/2$.
4. **(Chernoff the Bins.)** Suppose you throw m balls into n bins, each ball equally likely to go into any of the n bins; imagine $m \geq n$. Let r.v. B_i denote the number of balls in bin i. What is $E[B_1]$?

   (a) Suppose $m = 100n \ln n$. Use the multiplicative version of the Chernoff bounds to show that the number of balls in bin i does not differ from the expectation by more than (say) $25 \ln n$ with probability at least $1 - 1/n^2$. Hence, show that the load of the heaviest and lightest bins differ by at most a constant factor with probability at least $1 - 1/n$.

   (b) For general $m = \Omega(n \ln n)$, show that the number of balls in all the bins lie in the range $\frac{m}{n} \pm O(\sqrt{\frac{m}{n} \ln n})$ with probability at least $1 - 1/n$.

   (c) Now suppose $m = n$. Show that the height of the heaviest bin is $O(\frac{\ln n}{\ln \ln n})$ with probability $1 - o(1)$.

Problems

1. **(Generating random permutations.)** Here is an interesting way of producing a random permutation of a given list of n numbers. Run the simplest version of quicksort—using the leftmost element as pivot—but replacing the comparison operator with a random coin. That is, each time the algorithm asks "is $a < b$?", just flip a coin to give the answer. Additionally, after all elements have been compared to the pivot, put the pivot itself into one of the two buckets at random (so every element is now in one of the two buckets). Prove that this indeed yields a random permutation.

   Aside: Observe that we need at least $\log_2(n!) = n\log_2 n - \Theta(n)$ *bits of (unbiased) randomness to achieve a uniformly random permutatation; for each of the shufflers in this homework, it is worth thinking about how randomness-efficient the shuffler is. You don't have to hand in these observations.*

2. **(Hidden coverage.)** The NP-hard *set-cover* problem is: given subsets $S_1, S_2, \ldots, S_m$ of $[n] := \{1, 2, \ldots, n\}$, find the fewest subsets needed to cover all the points. Let k denote the number of sets used in an optimal solution; i.e., there exist k sets whose union is $[n]$.

   (a) Prove that the greedy algorithm (choose the set that covers the most new points) has the property that after k steps, it has found k sets covering at least a $(1 - 1/e)$ fraction of the points.

   (b) Now, suppose that we don't care about the total number of points covered and instead we only care about the number of *important* points covered. Unfortunately, we don't know which points are important! For example, imagine we have a security system with m cameras in an art museum, but only have funds to monitor k of the cameras; a thief is going to steal one of the n artworks according to some probability distribution $\vec{p} = (p_1, \ldots, p_n)$ but we don't know the distribution $\vec{p}$ and we want the highest probability of catching him.

   Thankfully, there exist k cameras that can be monitored to guard all n artworks—but we don't know them, and finding these k locations is NP-hard, so likely to take more time than we have. Describe a randomized (polynomial-time) algorithm with the property that for any $\vec{p}$, the algorithm has probability at least $1 - 1/e$ of catching the thief. Equivalently, for any (unknown) importance-weighting of the points, the expected total *weight* of points covered by the randomized algorithm is at least a $1 - 1/e$ fraction of the total weight.

   Hint: think of randomized rounding of a linear program.

3. **(Negative Correlation.)** The standard proof of the Chernoff bound showing concentration for the function $X = \sum_{i=1}^{n} X_i$ done in lecture assumed that the variables X_i are independent. Suppose we have a set of 0-1 variables $\{X_i\}_{i\in[n]}$ that satisfy the following "negative correlation" property:

$$\forall I \subseteq [n], \quad \Pr(\wedge_{i\in I}(X_i = 1)) \leq \prod_{i\in I} \Pr(X_i = 1). \tag{1}$$

   For instance, this property on $I = \{1, 2\}$ gives us that

$$\Pr(X_2 = 1 \mid X_1 = 1) = \frac{\Pr(X_1 = 1 \wedge X_2 = 1)}{\Pr(X_1 = 1)} \leq \Pr(X_2 = 1),$$

   that is, conditioning on $X_1 = 1$ (it is set "high") makes it less likely that $X_2 = 1$ and more likely it is set "low".

   (a) Suppose $\hat{X}_i$ is an r.v. with the same distribution as X_i (i.e., $\Pr(\hat{X}_i) = 1) = \Pr(X_i = 1)$), but the $\hat{X}_i$'s are all independent of each other. Let $\hat{X} = \sum_i \hat{X}_i$. For any $I \subseteq [n]$, show that

$$E[\prod_{i\in I} X_i] \leq E[\prod_{i\in I} \hat{X}_i].$$

   (b) Hence, for $k \geq 0$, show that $E[X^k] \leq E[\hat{X}^k]$, and hence that $E[e^{tX}] \leq E[e^{t\hat{X}}]$ for $t \geq 0$.

(c) Read the proof of the additive Chernoff bound in the Lecture #5 handout, and show how to prove the following variant: for $\{0,1\}$ random variables $\{X_i\}_{i\in[n]}$ satisfying property (1),

$$\Pr[X \geq \mu + n\delta] \leq \exp(-2\delta^2 n).$$

(d) (Optional.) Can you see where the proof breaks down if we want to prove the bound on the lower tail? Can you suggest a property similar to (1) that suffices to prove the bound on the lower tail?

(e) (Optional.) Read the proof of the multiplicative Chernoff bound from the MR book, and use (1) to prove a version for negatively correlated r.v.'s.

4. **(Optional: Another Shuffler.)** Suppose we start off with a deck with n cards, with card 1 at the top, and n at the bottom. At each step, we remove the top card from the deck, and insert it at a uniformly random place in the deck (chosen from one of the n possible places). Let T be the r.v. such that on the T^{th} step we pick up the card $n-1$ and insert it randomly into the pack. Calculate $E[T]$. Show that after time T, the deck is fully shuffled (i.e., each of the $n!$ permutations are equally likely), and before time T is it not.