CMSC 858C, Spring 2009: Ungraded Homework 1

Note: We will have *ungraded* homework assignments such as this one, as well as ones that will be graded. I will post the solutions for all the assignments some time after they are handed out. You will get the most out of this course if you do your best to solve all the homework problems (whether they are graded or not) by yourself. The suggested time period to finish this assignment is two weeks.

1. Recall the notions of distribution function and density function for continuous random variables. Let X be a random variable taking on values in $[0, a]$, for some finite positive a. Assuming that X has a density function, prove that $\mathbf{E}[X] = \int_0^a \Pr[X > y] \; dy$.

2. Recall the different "levels" at which inclusion-exclusion can be truncated: odd levels give upper bounds on the probability of a union, and even levels give lower bounds. Choose, say, odd levels. Is it necessarily true that we get better (i.e., lower) upper bounds as we proceed to higher odd levels? Prove, or give a counterexample.

3. Luby's $(\Delta + 1)$-coloring algorithm took $O(\log n)$ steps with high probability, but we ignored message contentions due to adjacent nodes communicating at the same time. We fix this issue in this problem. Let us assume that:

- time is discrete and is divided into equal-length time slots;
- all nodes know the value of Δ;
- a node can send a message to all its neighbors in one time slot (due to the broadcast nature of radio spectrum); however, if a node receives two or more messages (from different neighbors) at the same time step, or if a node receives message(s) during a time slot in which it transmits, then all these messages "collide" and will have to be retransmitted;
- there are no acknowledgments, and there is no way for a node to know for sure if its transmission to its neighbors at a particular time t, has been successfully received by all of its neighbors.

Show how contention resolution can be added to Luby's protocol, so that even in the above model, we are done within a total of $O(\Delta \log^2 n)$ time slots with high probability (say, with probability at least $1 - 1/n^2$).

4. Prove that in Luby's $(\Delta + 1)$-coloring algorithm, it is true that even if each node sleeps with probability zero in every iteration, the probability of success in a given iteration is at least $1/4$ for every yet-uncolored node. (This may be a challenging problem.)

5. Recall the definition of the complexity class $\mathcal{P}$: A language $\mathcal{L}$ is in $\mathcal{P}$ iff there exists a polynomial time algorithm $\mathcal{A}$ such that for all inputs x,

$$x \in \mathcal{L} \rightarrow \mathcal{A} \text{ accepts}$$

$$x \notin \mathcal{L} \rightarrow \mathcal{A} \text{ rejects}$$

For the class $\mathcal{RP}$ [Randomized Polynomial Time], we say that a language $\mathcal{L}$ is in $\mathcal{RP}$ iff there exists a randomized polynomial-time algorithm (i.e., a deterministic polynomial-time algorithm that can generate and use up to polynomially many random bits) $\mathcal{A}$ such that for all inputs x,

$$x \in \mathcal{L} \rightarrow \Pr[\mathcal{A} \text{ accepts}] \geq 1/2$$

$$x \notin \mathcal{L} \rightarrow \Pr[\mathcal{A} \text{ accepts}] = 0$$

These algorithms are called *Monte Carlo* algorithms; their running time is fixed, but their answers might not be correct. Also note that the probability is one-sided (asymmetric). If $\mathcal{A}$ accepts then $x \in \mathcal{L}$ for sure, but if $\mathcal{A}$ rejects then x may belong to $\mathcal{L}$. Also, it is easy to drive down the error probability: if $x \in \mathcal{L}$, then the probability that $\mathcal{A}$ rejects x in k independent invocations is clearly at most 2^{-k}.

We can naturally define the complementary class co-$\mathcal{RP}$: A language $\mathcal{L} \in$ co-$\mathcal{RP}$ iff $\exists$ a randomized polynomial time algorithm $\mathcal{A}$ such that for all inputs x,

$$x \in \mathcal{L} \rightarrow \Pr[\mathcal{A} \text{ accepts}] = 1$$

$$x \notin \mathcal{L} \rightarrow \Pr[\mathcal{A} \text{ accepts}] \leq 1/2$$

Next, $\mathcal{ZPP}$ [Zero-error Probabilistic Polynomial Time] consists of those languages which have algorithms whose expected runtime is polynomial and whenever they terminate, they give correct outputs. An example of such an algorithm is the Randomized Quicksort algorithm. Such algorithms are also known as *Las Vegas* algorithms.

Prove that $\mathcal{RP} \cap$ co-$\mathcal{RP} = \mathcal{ZPP}$.