COMP8601: Advanced Topics in Theoretical Computer Science
Homework 2 **Due Date:** 10 Oct 2013

Rules: Discussion of the problems is permitted, but writing the assignment together is not (i.e. you are not allowed to see the actual pages of another student).

In order to score out of 100, you only need to attempt 4 out of 5 questions. However, if you wish, you can attempt all 5 questions, and additional scores will be counted as extra credits.

1. (25 pt) **Graphs with No Short-Cycles.** In this question, we show the following result. For each $l \geq 3$, and $n \geq 2^{l+2}$, there exists a graph, with n vertices and no cycles of length l or less, that has $\Omega(n^{1+\frac{1}{l-1}})$ edges.

   (a) Consider the random graph $G_{n,p}$, where $p \geq \frac{2}{n}$. For $3 \leq i \leq l$, let Y_i be the number of length-i cycles in $G_{n,p}$. Compute $E[Y_i]$.

   (b) Let $Y := \sum_{3 \leq i \leq l} Y_i$. Show that $E[Y] \leq (np)^l$.

   (c) By choosing an appropriate value of p, prove that there exists an n-vertex graph, with no cycles of length l or less, that has $\Omega(n^{1+\frac{1}{l-1}})$ edges.

   (d) Derandomize the above procedure, i.e., give a deterministic algorithm that returns a graph with the desired properties. Analyze the running time of your algorithm.

2. (25 pt) **Better Approximation Ratio for Set Cover.** In the lecture notes, a randomized algorithm to achieve approximation ratio $4 \ln 4n$ is described. We show that if we increase the number of repetitions slightly, we can actually achieve a better ratio. We assume that the number of elements in $\mathcal{U}$ is large enough, say $n \geq 20$.

   (a) Suppose we repeat the Simple Rounding Procedure for $T := \ln n + \lambda \ln \ln n$ times, where $\lambda > 0$ is some number we determine later. Suppose $\widehat{C}$ is the index set of the cover returned from the repeated runs, and we want approximation ratio ρ, for some $\rho > 1$ which we want as small as possible. Let B_1 be the event that $|\widehat{C}| \geq \rho \cdot \mathsf{OPT}$. Show that if you set $\rho := \ln n + 2\lambda \ln \ln n + 2$, then $Pr[B_1] \leq 1 - \frac{1}{\ln n}$.

   (b) Let B_2 be the event that there is some element that is not covered by $\widehat{C}$. What value should λ take such that $Pr[B_2] \leq \frac{1}{(\ln n)^2}$?

   (c) What is the approximation ratio ρ (in terms of only n) of the resulting algorithm? Can you give an upper bound on the failure probability? Failure means either there is some element in $\mathcal{U}$ that is not covered or the cover $\widehat{C}$ is too large.

   (d) Given any arbitrary $0 < \delta < 1$, how can you obtain the same approximation ratio, but with failure probability at most δ? How many times in total do you have to run the Simple Rounding Procedure?

3. (25pt) **More on Chernoff Bound.**

   (a) **The Other Half of Chernoff.** Suppose $X_0, X_1, \ldots, X_{n-1}$ are independent $\{0,1\}$-random variables, each having expectation p. Let $Y := \sum_i X_i$ and $\mu := E[Y]$. Using the method of moment generating function, prove the following.
   For all $\epsilon > 0$, $Pr[Y - \mu \geq \epsilon\mu] \leq \left(\frac{e^\epsilon}{(1+\epsilon)^{1+\epsilon}}\right)^\mu$.

   (b) n **Balls into** n **Bins (Revisited).** Using the Chernoff Bound from the previous part, we can obtain a better bound for the balls and bins problem. Suppose n balls are thrown independently and uniformly at random into n bins. Let Y_1 be the number of balls in the first bin.

      (i) Find a number N in terms of n such that $Pr[Y_1 \geq N] \leq \frac{1}{n^2}$. Please give the exact form and do not use big O notation for this part of the question.
      (Hint: if you need to find a number W such that $W \ln W \geq \ln n$, try setting $W := \frac{\lambda \ln n}{\ln \ln n}$, for some constant $\lambda > 0$. You can also assume that n is large enough, say $n \geq 100$.)

      (ii) Show that with probability at least $1 - \frac{1}{n}$, no bin contains more than $\Theta(\frac{\log n}{\log \log n})$ balls.

4. (25 pt) **Integration by Sampling.** Suppose we are given an integrable function $f : [0,1] \to [0,M]$, and we wish to estimate the integral $I(f) := \int_0^1 f(x)dx$. We only have *black box access* to the function f: this means that we are given a box such that when we provide it with a real number x, the box returns the value $f(x)$. Moreover, we assume the real computation model. In particular, we assume that storing a real number takes constant space, and basic arithmetic and comparison operator ($\leq$) take constant time. Suppose we are also given a random number generator **Rand[0,1]** that returns a number uniformly at random from [0,1], and subsequent runs of **Rand[0,1]** are independent. The goal is to design an algorithm that given black box access to a function $f : [0,1] \to [0,M]$ and parameters $0 < \epsilon, \delta < 1$, return an estimate of $I(f)$ with additive error at most ϵ and failure probability at most δ.

   (a) Show that this is not achievable by a deterministic algorithm. In particular, show that for any deterministic algorithm A, there is some function f such that the algorithm A returns an answer with additive error $\frac{M}{2}$.
   (Hint: Any deterministic algorithm A should run for a finite number of steps. What happens when the algorithm A is run on the zero function $f \equiv 0$?)

   (b) Using the random generator **Rand[0,1]**, design a randomized algorithm to achieve the desired goal. Give the number of black box accesses to the function f and the number of accesses to **Rand[0,1]** used by your algorithm.
   (Hint: Given a random pair $(x,y) \in [0,1] \times [0,M]$, what is the probability that $y \leq f(x)$?)

5. (25 pt) **Estimating the (Unknown) Fraction of Red Balls.** Suppose a bag contains an unknown number of red balls (assume there is at least one red ball) and

you are only allowed to sample (with replacement) uniformly at random from the bag. Design an algorithm that, given $0 < \epsilon, \delta < 1$, with failure probability at most δ, returns an estimate of the fraction of red balls with **multiplicative** error at most ϵ, i.e., if the real fraction is p, the algorithm returns a number $\widehat{p}$ such that $|\widehat{p} - p| \leq \epsilon p$. Give the number of random samples used by your algorithm.