CS 761 Randomized Algorithms **Last Updated:** June 22, 2018
Homework 3 **Due Date: July 6 (11pm), 2018**

You are allowed to discuss with others but not allowed to use any references except the course notes and the books "Probability and Computing" and "Randomized Algorithms". Please list your collaborators for each question. This will not affect your marks. In any case, you must write your own solutions.

There are totally 70 marks, and the full mark is 60.

---

1. **Graph Drawing**

   (10 marks) A graph is *planar* if it can be drawn on the plane such that the edges do not intersect with each other. It is a well-known result that a simple planar graph with n vertices can have at most $3n-6$ edges. We say a graph G has intersecting number k if k is the maximum number such that any drawing of G on the plane has at least k pairs of edges intersecting. By the above result, a simple bound is that the intersecting number is at least t if the graph has at least $3n-6+t$ edges. Prove that the intersecting number is at least $m^3/(64n^2)$ for any simple graph with at least $m \geq 4n$ edges.

2. **Almost k-wise Independence**

   (15 marks) Let $X_1, \ldots, X_n \in \{0,1\}$ be n (not necessarily independent) random bits. We say that they are ϵ-almost k-wise independent if for any subset S of size k, we have $|\Pr(\cap_{i \in S}(X_i = b_i)) - 1/2^k| \leq \epsilon$ where $b_i \in \{0,1\}$ for $1 \leq i \leq n$.

   Show the existence of a sample space of size $O((2^k k \log n)/\epsilon^2)$ for ϵ-almost k-wise independent bits. That is, show that there exists a set of m n-bit strings $y^{(1)}, y^{(2)}, \ldots, y^{(m)} \in \{0,1\}^n$ where $m = O((2^k k \log n)/\epsilon^2)$, such that if we pick a uniform random n-bit string $y^{(j)}$ and set $X_i = y_i^{(j)}$ then the X_i are ϵ-almost k-wise independent bits.

   Assuming we are given these strings, explain how to use them to derandomize the color coding algorithm for the k-path problem in L03, and analyze the time complexity of the resulting deterministic algorithm.

3. **Triangles**

   (10 marks) Consider a graph in $G_{n,p}$ with $p = 1/n$. Let X be the number of triangles in the graph, where a triangle is a clique with three edges. Show that

   $$\Pr(X \geq 1) \leq 1/6$$

   and that

   $$\lim_{n \to \infty} \Pr(X \geq 1) \geq 1/7.$$

4. **Error Correcting Code**

   (20 marks) We will prove the existence of an error correcting code with *zero* decoding error probability. The model of the noisy channel is slightly different: if we send n bits through the channel, then we are guaranteed that at most pn bits will be flipped (but an "adversary" can decide to flip an arbitrary subset of at most pn bits). We would like to find an encoding function $f : \{0,1\}^m \to \{0,1\}^n$ so that m is as large as possible, while any arbitrary pn errors on the codeword can be tolerated for all m-bit messages.

   Consider the following coding scheme. We generate an $n \times m$ matrix A where each entry of A is 0 with probability 1/2 and 1 with probability 1/2. To encode a message x of m bits, we compute the codeword $y = Ax$ where we use arithmetic modulo two so that y is an n-bit string. We then send the codeword y to the receiver through the noisy channel.

   (a) Prove that the receiver can recover the message with probability one if and only if Ax has at least $2pn + 1$ nonzero bits for all $x \neq 0$.

   (b) Prove that for any $\epsilon > 0$, we can achieve $m \geq (1 - H(2p) - \epsilon)n$ while the receiver can always recover the message for large enough n.

   (c) Prove that for any $\epsilon > 0$, we *cannot* achieve $m \geq (1 - H(p) + \epsilon)n$ while the receiver can always recover the message for large enough n.

5. **Graph Coloring**

   (15 marks) Let $G = (V, E)$ be an undirected graph and suppose each $v \in V$ is associated with a set $S(v)$ of $32r$ colors, where $r \geq 1$. Suppose, in addition, that for each $v \in V$ and $c \in S(v)$ there are at most r neighbors u of v such that c lies in $S(u)$.

   Use local lemma to prove that there exists a proper coloring of G assigning to each vertex v a color from its class $S(v)$ such that, for any edge $(u, v) \in E$, the colors assigned to u and v are different.

   Furthermore, give a polynomial time randomized algorithm to find such a coloring.