CS395T: Pseudorandomness and Combinatorial Constructions — Problem Set No. 2

David Zuckerman — due: Feb. 27

1. (10 points) Recall that in class we gave a randomized algorithm for MAX CUT for which the expected size of the cut was $m/2$, where m is the number of edges.

   a) This fact by itself doesn't guarantee that a cut of size at least $m/2$ will be output with high probability. For an arbitrary even m, give an example of an integral random variable $Z \in [0, m]$ such that $\mathrm{E}[Z] = m/2$ but

   $$\Pr[Z \geq m/2] \leq 2/m.$$

   b) Nevertheless, with modest probability the algorithm outputs a cut with weight close to $m/2$. In particular, show that for any random variable $Z \in [0, m]$ such that $\mathrm{E}[Z] = m/2$,

   $$\Pr[Z \geq (1-\delta)m/2] \geq \delta/(1+\delta).$$

2. (15 points) The ith coordinate of a code is useless if all codewords have the same value in the ith coordinate. Let C be a linear binary code of length n with no useless coordinates.

   a) Show that the expected weight of a random codeword is at least $n/2$.

   b) Give a randomness-efficient version of the algorithm suggested by (a) which uses just $O(\log n + \log \epsilon^{-1})$ random bits and outputs a codeword with expected weight at least $(1-\epsilon)n/2$.

   c) Derandomize this completely and give a deterministic polynomial-time algorithm which outputs a codeword of weight at least $n/2$. Note that this algorithm can be parallelized, whereas previous deterministic algorithms could not.

3. (10 points) In the previous problem set, you gave an efficient probabilistic algorithm to test whether $A \times B = C$, where A, B, and C are $n \times n$ matrices with integral entries. One solution is to multiply by a uniformly random vector in $\{0,1\}^n$. Show that if instead, a random n-bit string from an ϵ-biased distribution is chosen, an error will be detected with probability at least $(1-\epsilon)/2$.

4. (15 points) For your choice of constant c, give an explicit construction of $f(n)$ graphs on n nodes such that at least one of the graphs does not contain a clique or an independent set of size $c \log n$. How small can you make $f(n)$? Note: $f(n) = n^{O(\log n)}$ is possible with what we did in class, and it is an open question to do better.

5. (10 points) A code of length n over $\mathbb{F}_q$ is perfect if, for some positive integer e, all words in $\mathbb{F}_q^n$ are within distance e of *exactly* one codeword.

   a) If C is a perfect $[n, k, d]_q$ code, give an exact expression for the ratio $q^n/|C|$ in terms of n, k, d, and q.

   b) Show that Reed-Solomon codes are not perfect.