CS395T: Pseudorandomness and Combinatorial Constructions — Problem Set No. 4

David Zuckerman — due: April 3

1. (15 points)

   a) Let $G = (V, E)$ be an expander graph (directed or undirected) on $n = |V|$ nodes, such that every set of size $n/2$ has at least $n/2 + \alpha n$ neighbors. For any k, let $S_1, \ldots, S_k$ be arbitrary subsets of V such that for all i, $|S_i| \geq (1 - \alpha)n$. Show that there exists a path $v_1, \ldots, v_k$ in G such that for all i, $v_i \in S_i$.

   b) Let $k = O(\log n)$ and $\epsilon > 0$ a constant. Give a deterministic polynomial-time algorithm that constructs a set of vectors $T \subseteq [n]^k$, $|T| = \text{poly}(n)$ with the property that for any $S_1, \ldots, S_k \subseteq [n]$ with each $|S_i| \geq \epsilon n$, there exists a vector $(v_1, v_2, \ldots, v_k) \in T$ such that for all i, $v_i \in S_i$.

2. (15 points) The Leftover Hash Lemma requires a hash family mapping $\{0,1\}^n$ to $\{0,1\}^m$ of size at least 2^n. Construct a family of functions of size polynomial in $2^m \cdot n$ for which you can prove a similar lemma. (Hint: use almost s-wise independence for an appropriate choice of s.)

3. (10 points) Alice and Bob together agreed on a uniformly random n-bit string $x = x_1 x_2 \ldots x_n$. Unfortunately, Eve was able to pick $n/2$ bit locations $i_1, \ldots, i_{n/2}$ and subsequently learn the value of $x_{i_1} \ldots x_{i_{n/2}}$ (so $i_1, \ldots, i_{n/2}$ do not depend on x). Alice and Bob do not know which bits Eve has learned. Moreover, Eve can hear any further conversations between Alice and Bob. Give a randomized protocol for Alice and Bob to agree on an $m = \Omega(n)$ bit string y, such that Eve has negligible information about y. In particular, for any set $S \subseteq \{0,1\}^m$, $|S| = 2^{m-1}$, the probability that Eve can guess whether $y \in S$ is at most $1/2 + 2^{-\Omega(n)}$. (This probability is over the randomized choices in the randomized protocol, as well as the randomness in the choice of x.)

4. (15 points) Give the best upper bound you can on the expected time for a random walk on the n-cycle to visit each vertex n times. The right answer is $O(n^2)$, though $O(n^2 \log n)$ is easier to show.