CS 225: Pseudorandomness	**Prof. Salil Vadhan**
Problem Set 3	
Assigned: Tue. Mar. 10, 2009	Due: Wed. Apr. 1, 2009(1 PM)

- Recall that your problem set solutions must be typed. You can email your solutions to `cs225-hw@eecs.harvard.edu`, or turn in it to MD138. You may write formulas or diagrams by hand. Aim for clarity and conciseness in your solutions, emphasizing the main ideas over low-level details.
- If you use LaTeX, please submit both the source (`.tex`) and the compiled file (`.ps`). Name your files `PS3-yourlastname`.
- Starred problems are extra credit.

Problem 1. (Near-Optimal Sampling) Describe an algorithm for Sampling that tosses $O(m+\log(1/\varepsilon)+\log(1/\delta))$ coins, makes $O((1/\varepsilon^2)\cdot\log(1/\delta))$ queries to a function $f:\{0,1\}^m \to [0,1]$, and estimates $\mu(f)$ to within $\pm\varepsilon$ with probability at least $1-\delta$. (Hint: use expander walks to generate coins for a pairwise-independent sampler, and compute the answer via a "median of averages".) It turns out that these bounds on the randomness and query complexities are each optimal up to constant factors.

Problem 4.3. (A "Constant-Sized" Expander)

1. Let $\mathbb{F}$ be a finite field. Consider a graph G with vertex set $\mathbb{F}^2$ and edge set $\{((a,b),(c,d)) : ac = b+d\}$. That is, we connect vertex (a,b) to all points on the line $y = ax - b$. Prove that G is $|\mathbb{F}|$-regular and $\lambda(G) \le 1/\sqrt{|\mathbb{F}|}$. (Hint: consider G^2.)
2. Show that if $|\mathbb{F}|$ is sufficiently large (but still constant), then by applying appropriate operations to G, we can obtain a base graph for the expander construction given in Section 4.3.3, i.e. a $(D^8, D, 7/8)$ graph for some constant D.

Problem 4.4. (The Replacement Product) Given a D_1-regular graph G_1 on N_1 vertices and a D_2-regular graph G_2 on D_1 vertices, consider the following graph $G_1 \circledR G_2$ on vertex set $[N_1]\times[D_1]$: vertex (u,i) is connected to (v,j) iff (a) $u = v$ and (i,j) is an edge in G_2, or (b) v is the i'th neighbor of u in G_1 and u is the j'th neighbor of v. That is, we "replace" each vertex v in G_1 with a copy of G_2, associating each edge incident to v with one vertex of G_2.

1. Prove that there is a function g such that if G_1 has spectral expansion γ_1 and G_2 has spectral expansion γ_2, then $G_1 \circledR G_2$ has spectral expansion $g(\gamma_1,\gamma_2,D_2) > 0$. (Hint: Note that $(G_1 \circledR G_2)^3$ has $G_1 Ⓩ G_2$ as a subgraph.)

2. Show how to convert an explicit construction of constant-degree (spectral) expanders into an explicit construction of degree 3 (spectral) expanders.

3. Prove that a dependence on D_2 in Part 1 is necessary by showing that $\gamma(G_1 \textcircled{r} G_2) = O(1/D_2)$ for sufficiently large N_1.

Problem 4.6. (Unbalanced Vertex Expanders and Data Structures) Let us consider a $(K, (1-\varepsilon)D)$ bipartite vertex expander G with N left vertices, M right vertices, and left degree D.

1. For a set S of left vertices, a $y \in N(S)$ is called a *unique neighbor* of S if y is incident to exactly one edge from S. Prove that every left-set S of size at most K has at least $(1-2\varepsilon)D|S|$ unique neighbors.

2. For a set S of size at most $K/2$, prove that at most $|S|/2$ vertices outside S have at least δD neighbors in $N(S)$, for $\delta = O(\varepsilon)$.

Now we'll see a beautiful application of such expanders to data structures. Suppose we want to store a small subset S of a large universe $[N]$ such that we can test membership in S by probing just 1 bit of our data structure. A trivial way to achieve this is to store the characteristic vector of S, but this requires N bits of storage. The hashing-based data structures mentioned in Section 3.5.3 only require storing $O(|S|)$ words, each of $O(\log N)$ bits, but testing membership requires reading an entire word (rather than just one bit.)

Our data structure will consist of M bits, which we think of as a $\{0,1\}$-assignment to the right vertices of our expander. This assignment will have the following property.

Property Π: For all left vertices x, all but a $\delta = O(\varepsilon)$ fraction of the neighbors of x are assigned the value $\chi_S(x)$ (where $\chi_S(x) = 1$ iff $x \in S$).

3. Show that if we store an assignment satisfying Property Π, then we can probabilistically test membership in S with error probability δ by reading just one bit of the data structure.

4. Show that an assignment satisfying Property Π exists provided $|S| \leq K/2$. (Hint: first assign 1 to all of S's neighbors and 0 to all its nonneighbors, then try to correct the errors.)

It turns out that the needed expanders exist with $M = O(K \log N)$ (for any constant ε), so the size of this data structure matches the hashing-based scheme while admitting 1-bit probes. However, note that such bipartite vertex expanders do *not* follow from explicit spectral expanders as given in Theorem 4.37, because the latter do not provide vertex expansion beyond $D/2$ nor do they yield highly imbalanced expanders (with $M \ll N$) as needed here. But later in the class, we will see how to explicitly construct expanders that are quite good for this application (specifically, with $M = K^{1.01} \cdot \mathrm{polylog} N$).

Problem 4.7. (Error Reduction For Free*) Show that if a problem has a **BPP** algorithm with constant error probability, then it has a **BPP** algorithm with error probability $1/n$ that uses *exactly* the same number of random bits.