Homework II, Topics in Theoretical Computer Science 2014

Due on Tuesday April 1 at 14:00 (send an email to ola.svensson@epfl.ch). Solutions to many homework problems, including problems on this set, are available on the Internet, either in exactly the same formulation or with some minor perturbation. It is *not acceptable* to copy such solutions. It is hard to make strict rules on what information from the Internet you may use and hence whenever in doubt contact Ola Svensson. You are, however, allowed to discuss problems in groups with up to three students, but solutions should be handed in individually and please write with whom you have collaborated.

We remark that on this problem set the first problem is about finding information and here you should feel free to use any source.

1 (35 pts) In this exercise, you are allowed to find the solution on the Internet[1] and then write the proof of the following statement in your own words:

Expander walks: Consider an n-vertex d-regular graph $G = (V, E)$ such that the second largest eigenvalue of its normalized adjacency matrix is at most $1 - \epsilon$ for some constant $\epsilon > 0$. Then there exists a constant k (dependent on ϵ) such that for any subset $S \subseteq V$ of vertices with $|S| \leq n/2$,

$$\Pr[\text{all vertices of a } k \log n \text{ length lazy random walk lie in } S] < 1/n.$$

A $k \log n$ length lazy random walk is obtained as follows: a starting vertex is chosen uniformly at random; then at each of the $k \log n$ steps, the walk goes to a random neighbor with probability $1/2$ or stays at the current vertex with probability $1/2$.

We remark that the above statement is very useful for saving random bits. Suppose that the set S consists of "bad" vertices and that G is a constant degree expander. If $|S| = n/2$ then a test that picks a vertex at random would find a bad vertex with probability $1/2$. Repeating this test by doing $\log n$ tests independently would increase the probability to find a bad vertex to $1 - 1/n$. However, it would use $\log^2 n$ random bits. By using the above statement, we can get the probability $1 - 1/n$ by only using $O(\log n)$ bits. First $\log n$ random bits to choose the start vertex then $O(k \log n)$ random bits to describe the steps of the walk. Indeed, note that each step of the random walk can be described using a constant number of random bits as each vertex has a constant number of neighbors (i.e., constant degree).

[1] A good source is "Computational Complexity, a modern approach" by Sanjeev Arora and Boaz Barak.

2 (35 pts, inspired from Johan Håstad's course "Theoreticians toolkit" at KTH)

Let us look at the graph given by the hypercube. It has $n = 2^m$ vertices with labels given by all strings in $\{0,1\}^m$ and two nodes are connected iff their labels differ in exactly one coordinate. Your task is to study this graph from an expander perspective. It is not constant degree, but fairly sparse in that the degree is $m = \log n$.

2a (10 pts) Estimate to what extent the hypercube is a node-expander. Construct a set S (of size at most $n/2$) such that it has very few neighbors outside S (fewer than $d|S|$ for any constant d).

2b (10 pts) Estimate to what extent the hypercube is an edge-expander. Try to find a set S (of size at most $n/2$) such that relatively few of the edges adjacent to S has its other endpoint in its complement. More specifically, for any size $n = 2^m$, define a set S such that $|S| \leq n/2$ and the number of edges crossing the cut $(S, \bar{S})$ is at most $|S|$.

2c (15 pts) Find the eigenvalues and eigenvectors of the adjacency matrix of the hypercube. Do this without trying to find information on the Internet. Instead start with small examples and then generalize to any n.

3 (30pts) Design and analyze an efficient (polynomial time) algorithm for the following problem:

Input: An n-vertex d-regular graph $G = (V, E)$ and a non-negative vector $x \in \mathbb{R}^{|V|}$ such that $|\{v \in V : x_v > 0\}| \leq 0.01n$.

Output: A subset $S \subseteq V$ of the vertices such that $|S| \leq 0.01n$ and $h(S) \leq \sqrt{2\delta}$ where

$$\delta := \frac{\sum_{ij} M_{ij}(x_i - x_j)^2}{\frac{1}{n}\sum_{i,j}(x_i - x_j)^2}, \qquad h(S) = \frac{E(S, \bar{S})}{d \min\{|S|, |\bar{S}|\}},$$

and M is the normalized adjacency matrix of G.

You should give a complete proof that your algorithm outputs a subset S of the vertices satisfying the above.