6.856 — Randomized Algorithms

David Karger

Handout #11, Mar. 31, 2011 — Homework 7, Due 4/6

1. **This problem should be done without collaboration.** In class we talked about how to count the number of satisfying assignments to a DNF, which is equivalent to estimating the probability of a satisfying assignment if variables get unbiased random assignments. Suppose that instead each variable gets set to true with probability $p < 1/2$. Explain how to estimate the probability of getting a satisfying assignment.

2. Consider a set S of vectors in d-dimensional space. Suppose that you sample each of those vectors independently with probability p. Prove that there are at most d/p vectors of S in expectation that are not spanned by your sample.

3. A *flow* in an undirected graph is a set of edge-disjoint paths from a *source* vertex s to a *sink* vertex t. The *value* of the flow is the number of edge disjoint paths. The *s-t maximum flow problem* aims to a flow of maximum value. This quantity turns out to be equal to the *s-t minimum cut value:* the minimum number of edges that must be removed from the graph in order to disconnect vertex s from vertex t. There is an *augmenting path algorithm* that, given an s-t flow of value v, finds an s-t flow of value $v + 1$ in $O(m)$ time on an m-edge graph, or else reports that v is the maximum flow.

   Consider any undirected graph with m edges, s-t maximum flow v, and minimum cut c:

   (a) Prove for any constant ϵ, an s-t cut of value at most $(1 + \epsilon)v$ can be found in $\tilde{O}(mv/c^2)$ time.

   (b) Prove that for any constant ϵ, a flow of value $(1 - \epsilon)$ can be found in $\tilde{O}(mv/c)$ time.

   (c) Sketch an algorithm that finds the maximum flow in $\tilde{O}(mv/\sqrt{c})$ time, and give a informal argument as to its correctness.

   (d) Use the algorithm of part (c) to improve the running times of the algorithms in parts (a) and (b)

4. In class we gave an (ϵ, δ)-FPRAS for estimating the probability of a graph G disconnecting when each edge fails with probability p. You will show how to generate a random disconnected version of G from this distribution.

(a) Explain how $\Pr[F \mid x_e]$ can be computed as a network reliability problem on a different graph, for both values of x_e.

(b) Let G be a graph and F the event that it fails. Let x_e be the state of a given edge (up or down). Give an FPRAS for computing $\Pr[x_e \mid F]$.

(c) Using self-reducibility, give an algorithm that produces a random disconnected version of G, conditioned on F.