CMSC 858S: Randomized Algorithms

Fall 2001

Ungraded Homework Assignment #1

Please note: We will have *ungraded* homework assignments such as this one, as well as ones that will be graded. I will post the solutions for all the assignments some time after they are handed out. You will get the most out of this course if you do your best to solve all the homework problems (whether they are graded or not) by yourself. The suggested time period to finish this assignment is 1 week.

1. Suppose we are given a polynomial p as a black-box, along with the guarantee that it has degree at most d; we want to know if p is identically zero or not. Suppose further that we are restricted to provide inputs from the set $\{0, 1, \ldots, n-1\}$ to p, where $n \gg d$. We saw in class an elementary *randomized* algorithm for the problem that gives just one input to p, and has an (one-sided) error probability of at most d/n. Suppose that we desire a *deterministic*, i.e., error-free algorithm. Show that $d+1$ inputs must be provided to p in the worst case.

Thus we see that at the cost of incurring a small uncertainty, randomization can give us tremendous gains over deterministic algorithms.

2. Do problem 1.2(a) in Motwani-Raghavan. Note how a problem such as this comes up, e.g., in the Random Quicksort algorithm.

3. Recall that events $A_1, A_2, \ldots, A_n$ are (pairwise) *mutually exclusive* if no two of them can occur simultaneously; i.e., if $i \neq j$, then $\Pr[A_i \wedge A_j] = 0$. As we saw in class, we have

$$\Pr[A_1 \vee A_2 \vee \cdots \vee A_n] = \sum_{i=1}^{n} \Pr[A_i]$$

in this case. This can be used sometimes to get some interesting bounds, as we see in this problem.

(a) Recall the analysis of Karger's random contraction algorithm for global min-cuts. Use the results of this analysis to show that any graph with n vertices can have at most $\binom{n}{2}$ global min-cuts.

(b) Suppose we have a finite set S of distinct nonempty binary strings (i.e., each string being composed of zeroes and ones); suppose further that no string is a prefix of any other (e.g., S cannot have both of the strings 010 and 010110). Suppose that for each $i \geq 1$, n_i is the number of strings in S of length i. Show that $\sum_{i \geq 1} n_i/2^i \leq 1$.
Hint: Use the fact that no string is a prefix of the other to construct a rooted binary tree T where each leaf corresponds to some distinct element of S. Now consider a "random walk" starting from the root of T, which is conducted as follows. We start at the root of T. Suppose in general that we are at some node u of T now. If u is a leaf, we stop; if u has only one child, we move to this child of u; else if u has two children, we choose one of these children uniformly at random, and move to this child. This random walk stops at some leaf X of T. Now what can you say about the distribution of this random variable X?

4. Suppose we are given an n-vertex undirected graph $G = (V, E)$, where the vertices are numbered $1, 2, \ldots, n$. We wish to find a (hopefully "large") independent set in G. Given a permutation σ of the vertices, define a subset $S(\sigma)$ of the vertices as follows: for each vertex i, $i \in S(\sigma)$ iff no neighbor j of i precedes i in the permutation σ.

(a) Show that each $S(\sigma)$ is an independent set in G, and that each independent set in G is a subset of $S(\sigma)$ for some permutation σ. Thus, if we are clever in finding a suitable choice of σ, then we may find a "large" independent set in G; this is what we explore next.

(b) Suggest a natural randomized algorithm to produce σ, for which you can show that the expected cardinality of $S(\sigma)$ is $\sum_{i=1}^{n} 1/(d_i + 1)$; here, d_i denotes the degree (i.e., the number of edges incident on) vertex i.

In particular, we see that G has an independent set of size at least $\sum_{i=1}^{n} 1/(d_i + 1)$.