

CS 7530 - Randomized Algorithms
Homework 1
Assigned Tuesday, August 31, 2004
Due Tuesday, Sept. 14, 2004

1. Describe a method for using a fair coin (a source of unbiased random bits) to generate a random permutation of $\{1, 2, \dots, n\}$, using a “quicksort” paradigm. Your method should use an expected $O(n \log n)$ random bits and take expected time $O(n \log n)$.

2. Consider adapting the min-cut algorithm to the problem of finding an $s-t$ min-cut in an undirected graph. In this problem, we are given an undirected graph G together with two distinguished vertices s and t . An $s-t$ min-cut is a set of edges whose removal disconnects s from t , and we seek an edge set of minimum cardinality.

As the algorithm proceeds, the vertex s may get amalgamated into a new vertex as the result of an edge being contracted; we call this vertex the s -vertex. Similarly, we have the t -vertex. We run the algorithm as before, except that we reject moves that would contract an edge between the s -vertex and the t -vertex.

(a) Show that there are simple graphs in which the probability that this algorithm finds an $s-t$ min-cut is exponentially small.

(b) How large can the number of $s-t$ min-cuts be in a single instance of G ?

3. Consider a uniform rooted tree of height h (i.e., every leaf is at distance h from the root). All internal vertices, including the root, have 3 children. Each leaf is assigned a boolean value (0 or 1). Each internal node returns the value returned by the majority of its children. The evaluation problem consists of determining the value of the root; at each step an algorithm can choose one leaf whose value it wishes to read.

(a) Show that for any deterministic algorithm, there is an instance (a set of boolean values for the leaves) that forces it to read all $n = 3^h$ leaves.

(b) Show that there is a nondeterministic algorithm that can determine the value of the tree by reading at most \sqrt{n} leaves. In other words, prove that one can present a set of this many leaves from which the value of the tree can be determined.

(c) Consider the recursive randomized algorithm that evaluates two subtrees of the root chosen at random. Only if the values disagree, it then proceeds to evaluate the third subtree. Show that the expected number of leaves read by the algorithm on any instance is at most $O(n^{0.9})$.

4. Suppose that you wish to evaluate the fraction f of republicans in Georgia. Assume that you are able to select a resident uniformly at random and determine their political affiliation. Assume also that you know some lower bound $a < f$. Devise a procedure for estimating f by some \hat{f} such that $\Pr[|f - \hat{f}| > \epsilon f] < \delta$, for every choice of constants $0 < a, \epsilon, \delta < 1$. Let N be the number of residents you must query to get the estimate. What is the smallest value of N for which you can give this guarantee? Use Chernoff bounds to get your answer.

5. We have a standard (fair) six-sided die. Let X be the number of times that a 6 occurs over n throws of the die. Let p be the probability of the event that $X \geq n/4$. Compare the best upper bounds on p that you can obtain using Markov's inequality, Chebyshev's inequality, and Chernoff bounds.

6. Consider the extension of the coupon collector's problem to that of collecting at least k copies of each coupon type. Find the expected number of coupon's that must be collected, and then prove a tail bound that generalizes what we did in class.

7. Show that Randomized Quicksort runs in time $O(n \log n)$ with high probability. (See exercise 20 in section 4.6 of Mitzenmacher and Upfal for an outline of how to approach this.)