Qualifying Examination

Theoretical Computer Science

Friday, February 15, 2008

Part I: Algorithms

ID Number	
Pseudonym	

Problem	Maximum Points	Points Earned	Grader
1	50		
2	50		
3	50		
4	50		
Total	200		

Instructions:

1. This is a closed book exam.
2. The exam is for 3 hours and has four problems of 50 points each. Read all the problems carefully to see the order in which you want to tackle them.
3. Write clearly and concisely. You may appeal to some standard algorithms/facts from text books unless the problem explicitly asks for a proof of that fact or the details of that algorithm.
4. If you cannot solve a problem, to get partial credit write down your main idea/approach in a clear and concise way. For example you can obtain a solution assuming a clearly stated lemma that you believe should be true but cannot prove during the exam. However, please do not write down a laundry list of half baked ideas just to get partial credit.

May the force be with you.

Problem 1. *Arithmetic coding* is a standard compression method. In the case when the string to be compressed is a sequence of biased coin flips, it can be described as follows. Suppose that we have a sequence of bits $X = (X_1, X_2, \ldots, X_n)$, where each X_i is independently 0 with probability p and 1 with probability $1 - p$. The sequences (which are n-bit zero-one vectors) can be ordered lexicographically; for $x = (x_1, x_2, \ldots, x_n)$ and $y = (y_1, y_2, \ldots, y_n)$, we say that $x < y$ if $x_i = 0$ and $y_i = 1$ in the first coordinate i such that $x_i \neq y_i$. If $z(x)$ is the number of zeroes in the string x, then define $p(x) = p^{z(x)}(1-p)^{n-z(x)}$ and

$$q(x) = \sum_{y<x} p(y).$$

(A) **[10 points]** Suppose we are given $x = (x_1, x_2, \ldots, x_n)$. Explain how to compute $q(x)$ in time $O(n)$ (assume that any reasonable operation on real numbers takes constant time).

(B) **[10 points]** Argue that the 2^n intervals $\Big[q(x), q(x) + p(x)\Big)$ defined by the set of all n-bit zero-one vectors x are disjoint subintervals of $[0, 1)$.

(C) **[15 points]** Given (A) and (B), the sequence X can be represented by any point in the interval $I(X) = \Big[q(X), q(X) + p(X)\Big)$. Show that we can choose a codeword in $I(X)$ with $\lceil \lg(1/p(X)) \rceil + 1$ binary decimal digits to represent X in such a way that no codeword is the prefix of any other codeword.

(D) **[15 points]** Given a codeword chosen as in (C), explain how to decompress it to determine the corresponding sequence $(X_1, X_2, \ldots, X_n)$.

(E) **[Bonus Points]** Using the Chernoff inequality, argue that $\lg(1/p(X))$ is close to $n\mathbb{H}(p)$ with high probability. Thus, this approach yields an effective compression scheme.

Reminder: Chernoff inequality states that for n random (and independent) variables $X_1, \ldots, X_n$, $Y = \sum_i X_i$ and $\mu = \mathbf{E}[Y]$, the following holds.

For $\delta \leq 2e - 1$	$\mathbf{Pr}[Y > (1+\delta)\mu] < \exp\big(-\mu\delta^2/4\big)$
$\delta \geq 2e - 1$	$\mathbf{Pr}[Y > (1+\delta)\mu] < 2^{-\mu(1+\delta)}$
For $\delta \geq 0$	$\mathbf{Pr}[Y < (1-\delta)\mu] < \exp\big(-\mu\delta^2/2\big)$

Also, for a random variable X its *entropy* is

$$\mathbb{H}(X) = \sum_x \mathbf{Pr}[X = x] \lg \frac{1}{\mathbf{Pr}[X = x]}.$$

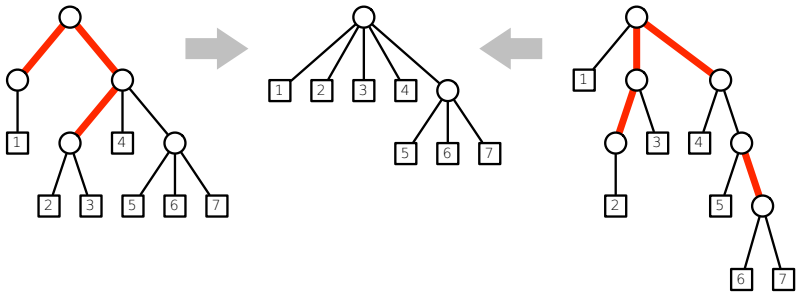
Problem 2. In this problem the goal is to develop a randomized algorithm that given a directed graph $G = (V, E)$ and an integer k, decides whether G has a simple path of length at least k; the length of a path P is the number of vertices in it including the end points. The algorithm's running time will be $O(c^k \text{poly}(n))$ for some constant c. The algorithm consists of two steps. In the first step the algorithm picks independently for each vertex $u \in V$ a random color from $\{1, 2, \ldots, k\}$ according to the uniform distribution. In the second step the algorithm finds a longest *multi-colored* path — that is, a path consisting of vertices with different colors.

(A) **[15 points]** Suppose P is a path of length k (contains k vertices) in G. What is the *precise* probability that the vertices in P get different colors in the first step. Simplify the expression to show that it is at least $1/{c_1}^k$ for some fixed constant $c_1 > 1$.

(B) **[25 points]** Suppose vertices of G are colored using $\{1, 2, \ldots, k\}$. Give an algorithm that finds the longest multi-colored path in G in $O(c_2^k \text{poly}(n))$ time where $c_2 > 1$ is another fixed constant; $c_2 = 4$ should suffice. *Hint: use dynamic programming.*.

(C) **[10 points]** Show, using (a) and (b) that there is a randomized algorithm that in time $O(c^k \text{poly}(n))$ time decides with *high probability* (that is, with probability at least $(1 - 1/n^2)$ say) whether G has a path of length k or not.

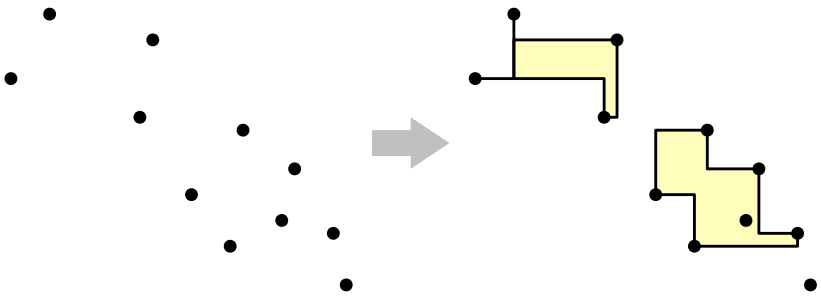
Problem 3. Let T be a rooted, ordered tree. An *internal edge* of T is an edge that is not incident to a leaf. A *contraction* of T is any tree obtained from T by contracting a subset of the internal edges. (See the figure below.) Because only internal edges are contracted, every contraction of T has the same number of leaves as T. A *common contraction* of two trees T_1 and T_2 is another tree that is both a contraction of T_1 and a contraction of T_2. Finally, we say that one tree is *larger* than another if it has more edges.



The middle tree is the largest common contraction of the left and right trees.

Describe and analyze an efficient algorithm to compute the *largest common contraction* of two given rooted, ordered trees. Assume both input trees have the same number of leaves.

Problem 4. A set of points in the plane is *orthogonally convex* if its intersection with any horizontal or vertical line is either a single line segment, a single point, or the empty set. The *orthogonal convex hull* of a set P of points is the smallest orthogonally convex set that contains P. Equivalently, the orthogonal convex hull is the complement of the union of all open axis-aligned quadrants (90° wedges) that do not intersect P. Unlike regular convex hulls, orthogonal convex hulls can be disconnected.



The orthogonal convex hull of these 11 points has three components.

Describe and analyze an algorithm to compute the orthogonal convex hull of a set of n points in the plane. For full credit, your algorithm should run in $O(n \log n)$ time. Be careful to explain exactly how your output is represented. You may assume that no two input points lie on a horizontal or vertical line.