Advanced Algorithms, Fall 2009	**Problem Set 3**
Instructor: Emanuele Viola	
Assigned: October 23	Due: November 6 before class

Guidelines If you write "I do not know how to solve this problem" then you get 1/4 of the score for the problem. If you write nonsense then you get 0.

As we are going to learn in this class, *time* and *space* are very valuable resources. Strive to give effective, compact solutions. Your solutions should touch on all the main points, but long calculations or discussions are not required nor sought.

Do not worry if you sometimes "do not get it." The problems are meant to stimulate you, not to overwork you. Unless specified otherwise, you can collaborate, but you must acknowledge all your collaborators in your solutions. I need your solutions on paper (not file). To hand them in: give it to me or slide them under my door West Village H (246).

Problem 1. Hash functions for long identifiers You need to hash L-bit identifiers. Your programmer is using very long identifiers (think $L = 10000$), and you are concerned because the hash functions seen in class have a description length at least L bits.

Give a family of hash functions mapping L-bit identifiers into $O(\log L)$ bits (= poly(L) elements) so that

(1) the description of a hash function requires only $O(\log L)$ bits, and

(2) for any two distinct identifiers, the probability, over the choice of the hash function, that their hashes coincide is at most $1/L$.

Hint: Interpret an identifier as a polynomial over a finite field F of suitable size. Use the fact that any non-zero polynomial $p(x)$ of degree d over a field F has at most d roots (i.e., elements x such that $p(x) = 0$), exactly as is the case over the real numbers.

You are still not satisfied, as you would like the range of the hash function to be $t \ll$ poly(L). Give a hash function with range t such that

(1') the description of a hash function still requires only $O(\log L)$ bits, and

(2') for any two distinct identifiers, the probability, over the choice of the hash function, that their hashes coincide is at most $1/L + 1/t$.

You can assume that L and t are powers of 2.

Problem 2. Universal hash functions without finite fields. Your are to develop a hash function to hash elements from $\{0,1\}^u$ into $\{0,1\}^r$ that satisfies the property P: for any two distinct identifiers, the probability, over the choice of the hash function, that their hashes coincide is at most 2^{-r}. However later you are to explain the construction to your Boss who never studied any math, so you cannot use the construction seen in class using finite fields.

(1) Consider the following family of functions for a picked randomly in $\{0,1\}^u$: $h_a(x) := \sum_{i=1}^{u} a_i \cdot x_i$ modulo 2. Show that this satisfies P for $r = 1$.

(2) Prove that if $h_a, h'_{a'}$ satisfy P, then the family (for random a, a') $(h_a, h'_{a'})(x) := (h_a(x), h'_{a'}(x))$ also satisfies P.

(3) Conclude that there is a solution to your hashing problem where the description of a hash function takes $u \cdot r$ bits. Verify that this description length is longer than that of the family seen in class.

(5) Now consider the following family based on convolution. For a random $a \in \{0,1\}^{r-1+u}$,

$$h_a(x) := (\sum_{i=1}^{u} a_{r-1+i} \cdot x_i, \sum_{i=1}^{u} a_{r-2+i} \cdot x_i, \ldots, \sum_{i=1}^{u} a_{1+i} \cdot x_i, \sum_{i=1}^{u} a_{0+i} \cdot x_i),$$

where the sum in each entry is modulo 2, and so the output of h is in $\{0,1\}^r$. Prove that this family satisfies P.

(6) Observe that the construction in (5) has description length close to the construction seen in class. (We may see later that this construction can be computed very efficiently.)

Problem 3. Log-space randomized algorithm with zero error for connectivity We have seen in class a randomized algorithm A that, given an undirected, unweighted graph G on n nodes and two nodes s, t, satisfies the following:

(1) It uses space $O(\log n)$,

(2) if s, t are not connected, it always says "NOT CONNECTED," while

(3) if s, t are connected, it says "CONNECTED" with probability at least $1/2$.

Use A to construct another randomized algorithm B for connectivity with the following properties:

(1') It uses space $O(\log n)$,

(2') on every input, B outputs "DON'T KNOW" with probability at most $1/2$, but

(3') whenever B does not say "DON'T KNOW," it always correctly computes whether s, t are connected or not.

Hint: First solve the problem assuming that the algorithm is given the total number P_n of pairs (u, v) in the graph that are connected. Then show how to compute P_n using an inductive approach similar to the one in the Floyd-Warshall algorithm seen in class.

You may find it helpful to first construct from A a new algorithm where (1-2) still hold but the error bound in (3) is $1/n^{2000}$ instead of $1/2$.

You may also find it helpful to use the fact that if the edge relation of another graph G' is computable from G in deterministic space $O(\log n)$, then one can simulate $A(G')$ with only a negligible overhead in space and time.