CS395T: Randomized Algorithms — Problem Set No. 2
David Zuckerman — due: Sep. 26, 2006

1. Exercise 1.2 (page 9).

2. Exercise 14.5 (page 398).

3. Problem 14.4.

4. Let p be prime. Suppose you have an efficient program P which purportedly computes some unknown linear function f over $\mathbb{Z}_p$. The program P may be faulty, but it is correct on at least 4/5 of the inputs. Give an efficient randomized algorithm to compute f which, for *each* input, is correct with probability at least .9.

5. Suppose you are given a randomized subroutine S which, on input (a, p), computes a square root of a mod p if p is prime and a is a square mod p. If a is not a square mod p or p is composite, then S may exhibit arbitrary behavior. You are also given a known upper bound $T(n)$ on the expected running time of S on any input (a, p) with p an n-bit prime. Give a randomized (BPP) primality test which runs in time $O(T(n) + A(n))$, where $A(n)$ upper bounds the time of an arithmetic operation (addition, subtraction, or multiplication). You may assume that the input is either prime or has at least two distinct prime factors.