Problem Set 4

This problem set is due **at 9:00pm** on **Wednesday, March 14, 2012**.

Both exercises and problems should be solved, but *only the problems* should be turned in. Exercises are intended to help you master the course material. Even though you should not turn in the exercise solutions, you are responsible for material covered by the exercises.

Mark the top of the first page of your solution with your name, the course number, the problem number, your recitation section, the date, and the names of any students with whom you collaborated. The homework template (LaTeX) is available on the course website.

You will often be called upon to "give an algorithm" to solve a certain problem. Your write-up should take the form of a short essay. A topic paragraph should summarize the problem you are solving and what your results are. The body of the essay should provide the following:

1. A description of the algorithm in English and, if helpful, pseudo-code.
2. At least one worked example or diagram to show more precisely how your algorithm works.
3. A proof (or indication) of the correctness of the algorithm.
4. An analysis of the running time of the algorithm.

Remember, your goal is to communicate. Full credit will be given only to correct solutions *that are described clearly*. Convoluted and opaque descriptions will receive lower marks.

Exercise 4-1. Do Exercise 5.2-5 on page 122 of CLRS.

Exercise 4-2. Do Exercise 5.3-4 on page 124 of CLRS.

Exercise 4-3. Do Exercise 7.4-1 on page 184 of CLRS.

Exercise 4-4. Do Exercise 7.4-5 on page 185 of CLRS.

Exercise 4-5. Do Exercise 9.2-2 on page 219 of CLRS.

Exercise 4-6. Do Exercise 9.3-3 on page 223 of CLRS.

Exercise 4-7. Do Exercise 9.3-8 on page 223 of CLRS.

Exercise 4-8. Do Exercise 25.2-6 on page 700 of CLRS.

Exercise 4-9. Do Exercise 25.3-1 on page 704 of CLRS.

Problem 4-1. Word-Search: Pattern Matching Revisited

In recitation we covered an $O(n \lg n)$ FFT-based algorithm for finding the offset in the text that gives the best match score between a pattern string and a target text. For this problem, we are interested in finding the first exact occurrence of the pattern in the text. Let $t = t_1 t_2 \ldots t_n$ be a target text and $p = p_1 p_2 \ldots p_m$ a pattern, both over alphabet $\Sigma = \{0, 1\}$ with $m \leq n$. Identifying the first exact occurrence of the pattern in the text amounts to finding the smallest $j \in \{1, 2, \ldots, n - m + 1\}$ such that for $1 \leq i \leq m$, it holds that $t_{j+i-1} = p_i$.

In this problem we will work with the word RAM model, where we can store each integer in a single word (or "byte") of computer memory. The number of bits that can be stored in a single word is $\lceil \lg n \rceil$. So, arithmetic operations on words or $O(\lg n)$-bit numbers take $O(1)$ time. Finally, with this model comparing two n-bit numbers takes $O(n)$ time.

For this problem you can assume that you are provided with a black-box prime number generator, and that it takes $O(1)$ time to sample $O(\lg K)$-bit primes, for K polynomial in n. Also, you can assume that $\pi(k) \sim \frac{k}{\ln k}$, where $\pi(k)$ be the number of distinct primes less than k.

(a) Define $f_p : \{0, 1\}^m \to \mathbb{Z}$ as $f_p(X) := g(X) \mod p$, where p is a prime and $g : \{0, 1\}^m \to \mathbb{Z}$ is a function that converts an m-bit binary string to a corresponding base 2 integer. Note that if X is equal to Y then $f_p(X) = f_p(Y)$. However, if X differs from Y then it can still be the case that $f_p(X) = f_p(Y)$. We will refer to cases where the results of evaluating the function on two different string inputs are equal as false positives.

Take the set $P = \{p_1, p_2 \ldots p_t\}$, where p_i are all primes less than some large integer K. Suppose we choose a prime p uniformly at random from the set P and take m-bit strings X and Y such that $X = Y$. Prove that we can bound the probability of a false positive as follows:

$$P\left(f_p(X) = f_p(Y)\right) \leq \frac{m}{t}$$

Hint: Consider the prime factorization of $|g(X) - g(Y)|$. Notice that the number of prime factors is at most m.

(b) Let $X(j)$ be a length-m substring of the target text that starts at position j, for a given $j \in \{1, 2, \ldots, n - m + 1\}$. Design a randomized algorithm that given $f_p(Y)$ and $f_p(X(j))$ determines if there is a match between the pattern and the target text for a given offset $j \in \{1, 2, \ldots, n - m + 1\}$.

(c) Design a formula that given $g(X(j))$ computes $g(X(j+1))$, where $X(j)$ is a length-m substring of the target text that starts at position j, for a given $j \in \{1, 2, \ldots, n-m+1\}$. Use it to compute $f_p(X(j + 1))$ from $f_p(X(j))$.

Note that the formula should depend on X, j, and m.

(d) Suppose that $X(j)$ and Y differ at every string position. Give the best upper bound you can on the expected number of positions j such that $f_p(X(j)) = f_p(Y)$, where $j \in \{1, 2, \ldots, n - m + 1\}$.

(e) Using parts above, design a randomized algorithm that determines if there is a match between a pattern and a target text in $O(n+m)$ expected running time. The algorithm should always return the correct answer.

(f) Provide a bound for the probability that the running time is more than 100 times the expected running time.