Assignment 11

CSC 513 April 5, 2018

Exercise 1: (50 points) A teacher has a problem: she is absolutely sure that a student has plagiarized some text on a recent book report. One of the student's sentences sounds oddly familiar, but the teacher can't quite figure out where it came from. The teacher decides to see if a smart CSC 513 student can help her out.

The teacher gives you a DVD containing the full text of the University library. The data is stored in a binary string $T[1],T[2], \ldots ,T[n]$, which we view as an array $T[1 \ldots n]$, where each $T[i]$ is either 0 or 1. She also gives you the quote from the student's book report, a shorter binary string $P[1 \ldots m]$, again where each $P[i]$ is either 0 or 1, and where $m < n$. For a binary string $A[1 \ldots k]$ and for integers i,j with $1 \leq i \leq j \leq k$, we use the notation $A[i \ldots j]$ to refer to the binary string $A[i]$, $A[i+1]$, $\ldots$, $A[j]$, called a substring of A. The goal of this problem is to determine whether P is a substring of T, i.e., whether $P = A[i \ldots j]$ for some i,j with $1 \leq i \leq j \leq n$.

For the purpose of this problem, assume that you can manipulate $\mathcal{O}(\log n)$-bit integers in constant time. For example, if $x \leq n^7$ and $y \leq n^5$, then you can calculate $x + y$ in constant time. On the other hand, you may not assume that m-bit integers can be manipulated in constant time, because m may be too large. For example, if $m = \Theta(log^2 n)$ and x and y are each m-bit integers, you cannot calculate $x + y$ in constant time. (In general, it is reasonable to assume that you can manipulate integers of length logarithmic in the input size in constant time, but larger integers require proportionally more time.)

a) Assume that you have a hash function $h(x)$ that computes a hash value of the m-bit binary string $x = A[i \ldots (i + m - 1)]$, for some binary string $A[1 \ldots k]$ and some $1 \leq i \leq k - m + 1$. Moreover, assume that the hash function is perfect: if $x \neq y$, then $h(x) \neq h(y)$. Assume that you can calculate the hash function in $\mathcal{O}(m)$ time. Show how to determine whether P is a substring of T in $\mathcal{O}(mn)$ time.

b) Consider the following family of hash functions h_p, parameterized by a prime number p in the range $[2, cn^4]$ for some constant $c > 0$:

$$h_p(x) = x (mod\ p).$$

Assume that p is chosen uniformly at random among all prime numbers in the range $[2, cn^4]$. For a fix i with $1 \leq i \leq n - m + 1$, and let $x = T[i \ldots (i + m - 1)]$. Show that, for an appropriate choice of c, if $x \neq P$, then

$$\Pr_p\{h_p(x) = h_p(P)\} \leq \frac{1}{n}.$$

Hint: Recall the following two number-theoretic facts: (1) an integer x has at most $\lg x$ prime factros; (2) the Prime Number Theorem: there are $\Theta(x/\lg x)$ prime numbers in the range $[2, x]$.

c) How long does it take to calculate $h_p(x)$, as defined in part (b)? Hint: Notice that x is an m-bit integer, and hence cannot be manipulated in constant time.

d) For $1 \leq i \leq n - m$, show how to calculate $h_p(A[i \ldots (i + m - 1)])$, as defined in part (b)?

e) Using the family of hash functions from part (b), derive an algorithm to determine whether P is a substring of T in $\mathcal{O}(n)$ expected time.

Exercise 2: (20 points) Consider a typical vending machine that dispenses a person's choice of candy after it has received a total of 30 cents in nickels (5 cents), dimes (10 cents), and quarters (25 cents). In this case, the automaton's alphabet consists of three different sizes of coins, a state of the machine is the total amount of money that has been received since the last candy was dispensed, the initial state is that of not having received any coins since the last candy was dispensed, and a final state is that of having received at least 30 cents.

a) Assuming that the machine is not designed to make change and will therefore merely consume any overpayment, define a finite deterministic automaton that models the vending machine.

b) Give a finite deterministic automaton for a machine that returns change.

Exercise 3: (30 points)

a) CLRS Exercise 32.4-1

b) CLRS Exercise 32.4-3

a) CLRS Exercise 32.4-5