Design and Analysis of Algorithms
Computer Science 45
Due: in class Wednesday, April 2, 1997

Neal Young

Problem Set 1

Warm up (don't hand in):

1. Read CLR section 33.3, do 33.3-2 and 33.3-4.
2. CLR 33.4-2

   hint: First prove that for any three integers a, b, and n, if n divides $a \times b$ and $\gcd(n, a) = 1$, then n divides b.

Hand in:

1. CLR 33.4-4

   Try to be careful to get the details right in this exercise.

   hint: Don't be fooled by the star. Remember long division on polynomials from high-school algebra? Use it to prove that if a is any number and $f(x)$ is any polynomial, then $f(x) = g(x) * (x - a) + c$, where g is a polynomial of lesser degree and c is some number. What can you say about c if $f(a) \equiv 0 \pmod{p}$?

2. (Better analysis of Rabin-Karp)

   CLR gives a "hand-waving" analysis of the expected time spent due to spurious hits when the pattern does not occur in the text. We'll improve this analysis.

   Recall that m is the number of digits in the pattern, n is the number of digits in the text, and q is the modulus used for the arithmetic.

   Suppose q is chosen uniformly at random from the primes less than some number N (to be determined later). (Later in the course we'll discuss how to choose such a q.) We want to know how big N has to be in order for the expected time spent checking spurious hits to be small.

   (a) Prove that any m-digit number has at most $4m$ prime factors.

   (b) Let a and b be any two m-digit numbers. Consider the event that $a \equiv b \pmod{q}$. Prove that, over all random choices of q, the probability of this event is

   $$O\left(\frac{m}{N/\ln N}\right)$$

   no matter what a and b are. hint: Use Theorem 33.37 on page 837 of CLR.

   (c) Assume for this part that Theorem 33.37 is exact.

   Suppose you want to implement the algorithm (with q chosen randomly as described above) so that the expected time spent on any text and pattern not in the text is linear.

   How big should N be?

   What does this analysis tell you about how big a pattern you can handle, and still do all the arithmetic on 4-byte words?

3. **extra credit.** CLR 34.2-3 (Extending Karp-Rabin to two-dimensional patterns)

   There may not be just one "right" answer. Describe the best solution you can. Discuss its merits and shortcomings.