Basic Algorithms — Fall 2014 — Problem Set 2
Due: Sept. 18

Notation: For a nonnegative integer $N$, we write $\text{len}(N)$ for the length of $N$ written in binary.

1. Compute $\gcd(588, 210)$ in two different ways: by finding the factorization of each number, and by using Euclid’s algorithm. Also, apply the Extended Euclidean algorithm to find integers $x$ and $y$ such that $588x + 210y = \gcd(588, 210)$.

2. Find the inverse of: $20 \mod 79$, $3 \mod 62$, $21 \mod 91$, $5 \mod 23$.

3. Consider an RSA key with $p = 17$, $q = 23$, $N = pq$, and $e = 3$. What value of $d$ should be used for the secret key? What is the encryption of the message $x = 41$?

4. Give a polynomial-time algorithm for computing $a^b \mod p$, given $a$, $b$, and prime $p$. That is, your algorithm should run in time bounded by a polynomial in $\text{len}(a) + \text{len}(b) + \text{len}(e) + \text{len}(p)$.

5. A positive integer $N$ is called a perfect power if it is of the form $a^b$, where $a$ and $b$ are integers greater than 1. Give an efficient algorithm that on input $N$, determines if $N$ is a perfect power, and if so, outputs the corresponding values $a$ and $b$. Your algorithm should run in time polynomial in the length of $N$, i.e., $O((\text{len}(N))^c)$ for some constant $c$.

   Hint: you might start with the simpler problem of determining whether $N$ is a perfect square, i.e., of the form $a^2$ for some integer $a$.

6. Alice has three friends with RSA public keys $(N_i, e_i)$ for $i = 1, \ldots, 3$. Moreover, we have $e_i = 3$ for $i = 1, \ldots, 3$. Suppose Alice sends an encryption of the same message $x \in \{0, \ldots, \min(N_1, N_2, N_3) - 1\}$ to her three friends, and Eve intercepts the corresponding ciphertexts $y_1, y_2, y_3$. Show that Eve can compute $x$ efficiently.

   Hint: use the Chinese Remainder Theorem (discussed in recitation), to compute the integer $y = x^3$, and then apply the algorithm from previous exercise; you may assume that the $N_i$’s are pairwise relatively prime.

The following is for honors students only.

H1. Consider again Euclid’s algorithm. In the recursive formulation given in class and in the text, if the inputs are $a$ and $b$, with $a \geq b \geq 0$, then the algorithm is invoked recursively with inputs

$$\langle r_0, r_1 \rangle, \langle r_1, r_2 \rangle, \ldots, \langle r_\ell, r_{\ell+1} \rangle,$$

where $r_0 = a$, $r_1 = b$, and for $i = 1, \ldots, \ell$, we have

$$r_{i-1} = r_i q_i + r_{i+1}, \quad (1)$$

where $q_i = \lfloor r_{i-1}/r_i \rfloor$. In addition, $r_{\ell+1} = 0$. The quantity $\ell$ represents the recursion depth, and also measures the number of division steps performed by the algorithm.

   (a) Show that $b \geq F_\ell$, where $F_\ell$ is the $\ell$th Fibonacci number, as defined in Exercise 6 from last week. Use the result from that exercise to get an upper bound on $\ell$ in terms of $b$. Is this bound better or worse than the one given in the text (at the very end of §1.2.3)?

   (b) The text argues that the running time of Euclid’s algorithm is $O((\text{len}(a))^3)$. Show that the running time is in fact $O(\text{len}(a) \text{len}(b))$. Hint: use the fact that the running time of the division step (1) is actually $O(\text{len}(r_i) \text{len}(q_i))$.  