Takehome Project 1 Solutions

Written Part

6. (a) Prove that `millerRabin(a,n)` runs in $\mathcal{O}(\log n)$ basic operations $(+, -, \times, \div)$.

Proof. There are four main peices.

- As a first step the algorithm computes $\gcd(a, n)$ using the Extended Euclidean algorithm, which takes $\mathcal{O}(\log n)$ steps (this was not strictly necessary but since we are testing whether n is composite this may avoid extra computations).
- Then n is factored as $2^k \cdot m$, where m is odd. If n is a power of two, $k = \log_2 n$. Otherwise it is smaller, so we see that $k = \mathcal{O}(\log n)$. Factoring n as $2^k \cdot m$ involves dividing by 2 exactly $k = \mathcal{O}(\log n)$ times.
- As a third step we raise a to the power of m using fast powering, which runs in $\mathcal{O}(\log m)$ steps. Notice that $m \leq n$, so this step is in fact $\mathcal{O}(\log(n))$.
- Finally a^m is squared and reduced mod n a maximum of $k = \mathcal{O}(\log n)$ times (at each time checking if the result is -1).

In total we have four successive $\mathcal{O}(\log n)$ computations so the entire algorithm runs in $\mathcal{O}(\log n)$. □

(b) Assume that `ZZ.random_element` runs in $\mathcal{O}(1)$. Prove that `probablyPrime(n)` runs in $\mathcal{O}(\log n)$ basic operations.

Proof. `probablyPrime(n)` first computes a random element a (taking $\mathcal{O}(1)$), and then runs `millerRabin(a,n)` (taking $\mathcal{O}(\log n)$ by part (a)). It repeates this process 20 times, for a grand total of $\mathcal{O}(20 \log n) = \mathcal{O}(\log n)$. □

It is rather amazing that our probabilistic primality test runs so quickly. It is slightly more subtle to analyze the time complexity of `findPrime` (and therefore anything that calls it), since it relies on randomly guessing numbers before checking if they are prime. Part of the input to this is knowing the distribution of the primes which we haven't discussed yet. Nevertheless, we can discuss some of this.

(c) Describe how `findPrime` could possibly go on forever.
`findPrime` randomly picks a number an then runs the probably prime algorithm to check if it is prime. Although very unlikely, it is possible that the random number picked could repeatedly not be prime—for example, the same even number could theoretically be picked over and over again for ever.

(d) Show that it is extremely unlikely that `findPrime` runs forever. (*Hint*: For the second part, let P be proportion of primes in your range. To not pick a nonprime then has probabilitiy $1 - P$. Relate the problem to $\lim_{n\to\infty}(1 - P)^n$).

Proof. Suppose we run `findPrime(low,high)`. We let P be the proportion of primes between `low` and `high` (that is, $P = \frac{\pi(high)-\pi(low)}{high-low}$). Then the probability to pick a composite number at random is $1 - P$. If $P > 0$ then the probability of picking n nonprimes in a row is $(1-P)^n$. In particular, since $0 \geq (1-P)^n < 1$, we have $\lim_n \to \infty(1-P)^n = 0$.

In particular, as n grows the probability of picking n composite numbers in a row approaches 0, and so it is essentially inevatible that we will eventually pick a prime.

A caveat. For this argument to work we must know there is at least 1 prime between `low` and `high`. For example, suppose we ran `findPrime(14,17)`. Since $14, 15, 16$ are not prime, we would never be able to find a prime number. By the prime number theorem there should be around $\frac{high}{\ln(high)} - \frac{low}{\ln(low)}$ primes between `high` and `low`, so if we pick a wide enough range this should not be an issue. □

7. ~~Recall that if n is composite, then 75% of the integers between 2 and $n-1$ serve as Miller-Rabin witnesses to the compositness of n. Use this fact to compute the probability that n is prime if probablyPrime(n) returns true. Fully justify your answer.~~

8. We finish by proving the general case of Euler's theorem, and using this to show that the generalized RSA problem of solving $x^e \equiv c \mod N$ for x can be solved by factoring N. First let's review the following definition.

Definition 1 (Euler's φ-function)**.** *Let n be a positive integer. Then $\varphi(n)$ is the number of positive integers less than n which are coprime with n. That is:*

$$\varphi(n) = \#\{a = 1, \cdots, n-1 \text{ such that } \gcd(a, n) = 1\}.$$

(a) Show that $(\mathbb{Z}/n\mathbb{Z})^*$ is a finite commutative group under multiplication. What is its order?

Proof. If a, b are coprime to n, then so is ab, so that it is closed under multiplication. Since $\gcd(1, n) = 1$, we know that $1 \in (\mathbb{Z}/n\mathbb{Z})^*$ which is a multiplicative identity. Furthermore, every element prime to n has an inverse mod n by the extended Euclidean algorithm. Associativity and commutativity is inherited from $\mathbb{Z}$ it it is a commutative group! Finally, since every element of $\mathbb{Z}/n\mathbb{Z}$ can be represented by a residue between 0 and $n-1$, we see that elements of group of units are in bijection with numbers $a = 1, \cdots, n-1$ such that $\gcd(a, n) = 1$. Therefore the order is precisely $\varphi(n)$. □

(b) Prove Euler's Theorem: Let a be an integer coprime with n, then

$$a^{\varphi(n)} \equiv 1 \mod n.$$

(*Hint*: Recall that we proved Lagrange's theorem for commutative groups. Can this do all the work for you?).

Proof. Lagrange's theorem for finite commutative groups says that if G is a commutative group of order N, then for all $g \in G$ we have $g^N = 1_G$ (this is also true for noncommutative groups, but we only proved it in the commutative case in class). This immediately implies Euler's theorem with $G = (\mathbb{Z}/n\mathbb{Z})^*$ which has order $\varphi(n)$. □

(c) The Euler φ function has nice properties with respect to multiplication. Let's establish them:

i. Let m, n be positive coprime integers. Prove that $\varphi(mn) = \varphi(m)\varphi(n)$.

Proof. We give a bijection between elements of $(\mathbb{Z}/mn\mathbb{Z})^*$ and pairs of elements of elements in $(\mathbb{Z}/m\mathbb{Z})^*$ and $(\mathbb{Z}/n\mathbb{Z})^*$, which immediately implies the result. We first establish this fact without the $*$'s, noticing that it is essentially just a restatement of Sun Tzu's theorem.

Theorem 1 (Sun Tzu's Theorem: Ring Theoretic Statement)**.** *Let m, n be positive coprime integers. The reduction map*

$$\begin{array}{rcl} \mathbb{Z}/mn\mathbb{Z} & \to & \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z} \\ a & \mapsto & (a,a) \end{array}$$

is a bijection which respects multiplication and addition. (Although we don't have the language yet for this, this means it is an isomorphism of rings.)

Proof. First we show injectivity. Suppose a and b have the same image. This says that $a \equiv b \mod n$ and $a \equiv b \mod m$. The uniqueness part of Sun Tzu's theorem says that $a \equiv b \mod mn$. For surjectivity we choose $c_1 \in \mathbb{Z}/n\mathbb{Z}$ and $c_2 \in \mathbb{Z}/m\mathbb{Z}$. The congruences $x \equiv c_1 \mod n$ and $x \equiv c_2 \mod m$ have a solution $a \in \mathbb{Z}$, whose residue modulo mn maps to (c_1, c_2) under the reduction map. The fact that the reduction map respects multiplication and addition is HW1 Problem 9. □

We now must show that the reduction map restricts to a bijection on multiplicative groups. Explicitly, we must show that the residue of an integer a mod mn is in $(\mathbb{Z}/nm\mathbb{Z})^*$ if and only if its residues mod n and m are in $(\mathbb{Z}/n\mathbb{Z})^*$ and $(\mathbb{Z}/m\mathbb{Z})^*$ respectively. But this is immediate, as a is coprime with mn if and only if it shares no prime factors with mn, if and only if it shares no prime factors with m or n, if and only if it is prime to both m and n. Therefore we have shown that the reduction map gives a bijection

$$(\mathbb{Z}/mn\mathbb{Z})^* \leftrightarrow (\mathbb{Z}/m\mathbb{Z})^* \times (\mathbb{Z}/n\mathbb{Z})^*.$$

Since the left side has $\varphi(mn)$ elements, and the right side has $\varphi(m)\varphi(n)$ elements, we are done. □

ii. Let p be a prime number, and j a positive integer. Compute $\varphi(p^j)$. Justify your answer.

Proof. The numbers $\leq p^j$ which are *not* coprime to p^j are the multiples of p:

$$\{p, 2p, 3p, ..., (p^{j-1}-1)p, p^{j-1}p = p^j\}.$$

There are exactly p^{j-1} of these. $\varphi(p^j)$ counts the numbers $\leq p^j$ which are not in this set, so there are $p^j - p^{j-1}$ of these, so $\varphi(p^j) = \varphi(p^{j-1})$. □

iii. Using parts (c)i,ii prove the following formula for φ for a general integer $N > 1$:

$$\varphi(N) = N \cdot \left(\prod_{\substack{\text{primes } p \\ \text{with } p|N}} \left(1 - \frac{1}{p}\right) \right).$$

Proof. This follows from a formal manipulation. First let $N = p_1^{\alpha_1} \cdots p_t^{\alpha_t}$ be the prime factorization of N. Then we have:

$$\begin{aligned}
\varphi(N) &= \varphi\left(\prod_{i=1}^{t} p_i^{\alpha_i}\right) \\
&= \prod_{i=1}^{t} \varphi(p_i^{\alpha_i}) && \text{(Part i.)} \\
&= \prod_{i=1}^{t} \left(p_i^{\alpha_i} - p_i^{\alpha_i - 1}\right) && \text{(Part ii.)} \\
&= \prod_{i=1}^{t} \left(p_i^{\alpha_i}\left(1 - \frac{1}{p_i}\right)\right) && \text{(factoring)} \\
&= p_1^{\alpha_1} \cdots p_t^{\alpha_t} \left(\prod_{i=1}^{t}\left(1 - \frac{1}{p_i}\right)\right) && \text{(rearranging)} \\
&= N\left(\prod_{i=1}^{t}\left(1 - \frac{1}{p_i}\right)\right)
\end{aligned}$$

as desired. □

(d) Euler's theorem is related to several theorems we have already seen.

i. Deduce Fermat's little theorem from Euler's theorem.

Proof. Euler's theorem says for any a prime to p $a^{\varphi(p)} \equiv 1 \mod p$. Since $\varphi(p) = p-1$ we are done. □

ii. Let $N = pq$ be a product of distinct primes. Compare the formula given by Euler's theorem to the version of Euler's theorem for the product of two primes that we proved in class. Which one is stronger? Does one imply the other?

Proof. By part (c)i we know $\varphi(pq) = \varphi(p)\varphi(q) = (p-1)(q-1)$. Then Euler's theorem say any a prime to pq satisfies $a^{(p-1)(q-1)} \equiv 1 \mod pq$. In class we showed the formula $a^{(p-1)(q-1)/g} \equiv 1 \mod pq$, where g is any common divisor of $p-1$ and $q-1$. This second formula is stronger, as it implies the first one setting $g = 1$. □

(e) Fermat's little theorem helped us compute roots mod p, and Euler's theorem for the product of two primes helped us compute roots mod pq. Similarly, the general version of Euler's theorem allows us to compute roots mod N for general N. To see this fix N, c, e positive integers such that $\gcd(e, \varphi(N)) = 1$ and such that $\gcd(c, N) = 1$.

i. Show that $x^e \equiv c \mod N$ has a unique solution in $\mathbb{Z}/N\mathbb{Z}$.

Proof. Since $\gcd(e, \varphi(N)) = 1$, then e has a unique inverse d such that $ed \equiv 1 \mod \varphi(N)$ By Euler's theorem, for any integer z we have:

$$z^{ed} = z^{k\varphi(N)+1} = z(z^{\varphi(N)})^k \equiv z \mod N.$$

Then letting $x = c^d$ we see that:

$$x^e = c^{de} = \equiv c \mod N.$$

This proves the existence of a e'th root. To prove uniqueness, let y be another e'th root. Then:

$$y \equiv y^{ed} = (y^e)^d = c^d = x,$$

proving uniqueness. □

ii. Suppose you know the factorization of N. Describe an algorithm to compute the unique solution $x = \sqrt[e]{c} \mod N$ from part (e)i.

We compute the root in the following steps.

(1) Compute $\varphi(N)$ using the formula given in (c)iii. (This takes knowing the factorization of N.)

(2) Compute the inverse d of e modulo $\varphi(N)$ using the extended euclidean algorithm.

(3) Compute $x = c^d$ using the fast powering algorithm.

The correctness of this algorithm is part (e)i. Time complexity was not asked for, but is interesting: step 1 takes $\mathcal{O}(\omega(N))$ where $\omega(N)$ is the number of distinct prime factors of N. Steps 2 and 3 take $\mathcal{O}(\log \varphi(N))$ each. It is a theorem that $\omega(N) = \mathcal{O}(\log \log N)$. On the other hand, it is known that $\varphi(N)$ is essentally linear (that is, fix any $\varepsilon > 0$, then for large enough N we have $N^{1-\varepsilon} \leq \varphi(N) \leq N$), steps 2 and 3 are $\mathcal{O}(\log N)$ (in fact they are slightly faster). Steps 2 and 3 dominate, and the entire algorithm runs in $\mathcal{O}(\log N)$.