5 Probability in Computing

5.1 A Monte-Carlo Primality Tester

The factoring problem seems very hard. But to test if a number has factors turns out to be much easier than to find them. It also helps if we supply the computer with a coin-flipping device. See: [Miller 76, Solovay, Strassen 77, Rabin 80]. We now consider a Monte Carlo algorithm, i.e. one that with high probability rejects any composite number, but never a prime.

Residue Arithmetic. $p|x$ means p divides x. $x \equiv y \pmod p$ means $p|(x{-}y)$. $y = (x \bmod p)$ denotes the residue of x when divided by p, i.e. $x \equiv y \in [0, p{-}1]$. Residues can be added, multiplied and subtracted with the result put back in the range $[0, p{-}1]$ via shifting by an appropriate multiple of p. E.g., $-x$ means $p{-}x$ for residues mod p. We use $\pm x$ to mean either x or $-x$.

The Euclidean Algorithm finds $\gcd(x, y)$ – the greatest (and divisible by any other) common divisor of x and y: $\gcd(x, 0) = x$; $\gcd(x, y) = \gcd(y, (x \bmod y))$, for $y > 0$. By induction, $g{=}\gcd(x, y){=}A * x{-}B * y$, where integers $A{=}(g/x \bmod y)$ and $B{=}(g/y \bmod x)$ are produced as a byproduct of that algorithm. This allows division (mod p) by any r ***coprime*** with p, (i.e. $\gcd(r, p){=}1$), and operations $+, -, *, /$ obey all usual arithmetical laws. We also need to compute $(x^q \bmod p)$ in polynomial time. We cannot do $q{>}2^{\|q\|}$ multiplications. Instead we compute all numbers $x_i = (x_{i-1}^2 \bmod p) = (x^{2^i} \bmod p), i < \|q\|$. Then we represent q in binary, i.e. as a sum of powers of 2 and multiply mod p the needed x_i's.

Fermat Test. The Little Fermat Theorem for each prime $p \nmid x$ says: $x^{(p-1)} \equiv 1 \pmod p$. Indeed, the sequence $(xi \bmod p)$ is a permutation of $[1, p{-}1]$. So, $1{\equiv}(\prod_{i<p}(xi))/(p{-}1)! \equiv x^{p-1} (\bmod p)$. This test rejects typical composite p, including all $p{=}a^2b{\neq}b$: $(1{+}p/a)^{p-1}{=}1{+}(p/a)(p{-}1){+}(p/a)^2c \equiv 1{-}p/a \not\equiv 1 \pmod p$. Other composite p (***Carmichael numbers***) can be actually factored by the following tests.

Square Root Test. For each y and prime p, $x^2 \equiv y \pmod p$ has at most one pair of solutions $\pm x$.

Proof. Let x, x' be two solutions: $y \equiv x^2 \equiv x'^2 \pmod p$. Then $x^2 - {x'}^2 = (x{-}x')(x{+}x') \equiv 0 \pmod p$. So, $p \,|\, (x{-}x')(x{+}x')$. Thus p, if prime, divides either $(x{-}x')$ or $(x{+}x')$, making $x \equiv \pm x'$. Otherwise p is composite, and $\gcd(p, x{+}x')$ *actually gives* its factor.

Random Choice. We say d ***kills*** $\mathbb{Z}_p^*$ if $x^d \equiv 1 \pmod p$ for all x (in $\mathbb{Z}_p^*$, i.e. coprime with p). If $x^d \not\equiv 1$, then for all y either $y^d \not\equiv 1$ or $(xy)^d \not\equiv 1$. Same with $x^d \not\equiv \pm 1 \pmod p$. So existence of a single such x implies the same for **most** of randomly chosen y.

Miller-Rabin Test T_x factors a composite p given d that kills $\mathbb{Z}_p^*$. If $d = p{-}1$ does not, then Fermat Test confirms p is composite. Let $d{=}2^k q$, with odd q. T_x sets $x_0 = (x^q \bmod p)$, $x_i = (x_{i-1}^2 \bmod p) = (x^{2^i q} \bmod p)$, $i{\le}k$. $x_k{=}1$. If $x_0{=}1$, or one of x_i is -1, T_x gives up for this x. Otherwise $x_i \not\equiv \pm 1$ for some $i{<}k$, while $x_i^2 \equiv x_{i+1} \equiv 1$, and the Square Root Test factors p.

Now, for any coprime a, b, $p{=}ab$, T succeeds with some (thus most!) $x{\in}\mathbb{Z}_p^*$: Take the **greatest** i such that $2^i q$ does not kill $\mathbb{Z}_p^*$. It exists (as $(-1)^q \equiv -1$ for odd q) and has $x_i \not\equiv 1 \equiv (x_i)^2 \pmod p$. Then $x'{=}1{+}b(1/b \bmod a)(x{-}1) \equiv 1{\equiv}x_i' \pmod b$, while $x_i'{\equiv}x_i \not\equiv 1 \pmod a$. So, $x_i' \not\equiv \pm 1 \pmod p$.