Discrete Math, Thirteenth Problem Set (July 16)

REU 2003

Instructor: László Babai
Scribe: Daniel Štefankovič

1 Solovay-Strassen primality testing algorithm

Let $\mathbb{Z}_n^*$ be the set of integers $1 \le a \le n$ such that $\gcd(a, x) = 1$. Recall that the Euler phi function is defined by $\varphi(n) = |\mathbb{Z}_n^*|$.

Exercise 1.1. Prove that $\mathbb{Z}_n^*$ is a group under multiplication mod n.

Exercise 1.2. If there is $a \in \mathbb{Z}_n^*$ such that

$$a^{n-1} \not\equiv 1 \pmod{n} \tag{1}$$

then at least half of the numbers in $\mathbb{Z}_n^*$ satisfy (1).

Definition 1.3. Let p be an odd prime. For $a \in \mathbb{Z}$, the **Legendre symbol** $\left(\frac{a}{p}\right)$ is defined as follows.

$$\left(\frac{a}{p}\right) = \begin{cases} 0 & \text{if } p \mid a \\ 1 & \text{if } p \nmid a \text{ and } (\exists x)(x^2 \equiv a \pmod{p}) \\ -1 & \text{otherwise.} \end{cases}$$

In the second case we say that a is a **quadratic residue mod** p; in the third case, a is a **nonresidue mod** p. Note that 0 is *not* a quadratic residue even though $0^2 = 0$.

Theorem 1.4. *Let p, q be odd primes. The Legendre symbol satisfies the following identities.*

(1) If $a \equiv b \pmod{p}$ then $\left(\frac{a}{p}\right) = \left(\frac{b}{p}\right)$.

(2) $\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right)$;

(3) $\left(\frac{-1}{p}\right) = (-1)^{(p-1)/2}$;

(4) $\left(\frac{2}{p}\right) = (-1)^{(p^2-1)/8}$*;*

(5) $\left(\frac{p}{q}\right) = \left(\frac{q}{p}\right)(-1)^{\frac{p-1}{2}\cdot\frac{q-1}{2}}$ *(this identity, observed by Euler and Legendre and proved by Gauss, is called* Quadratic Reciprocity*);*

Exercise 1.5. Prove parts (1), (2), and (3) of Theorem 1.4.

Exercise 1.6. For what primes p is 5 a quadratic residue mod p ? What about 7?

Exercise 1.7. Show that using a factorization oracle (a black box that factors integers) and Theorem 1.4 we can compute the Legendre symbol in polynomial time. Note, however, that factoring is not expected to be doable in polynomial time.

Definition 1.8. Let a be an integer and b be an odd integer. Let $b = \prod_{i=1}^{\ell} p_i^{k_i}$ be the factorization of b. The Jacobi symbol $\left(\frac{a}{b}\right)$ is defined as follows.

$$\left(\frac{a}{b}\right) = \prod_{i=1}^{\ell} \left(\frac{a}{p_i}\right)^{k_i},$$

where the right hand side of the definition uses the Legendre symbol.

Exercise 1.9. Show that Theorem 1.4 holds for any odd p, q (not necessarily prime) with the Legendre symbol replaced by the Jacobi symbol.

Recall that Euclid's algorithm computes the greatest common divisor of integers a, b, using

$$\gcd(a, b) = \gcd(b \bmod a, a). \tag{2}$$

Exercise 1.10. Let $0 < a \leq b$. Let $a_1 = b \bmod a$, $b_1 = a$ and $a_2 = b_1 \bmod a_1$, $b_2 = a_1$. Show $|a_2| \leq |a|/2$. Conclude that the Euclid's algorithm terminates in $\leq 2n$ rounds, where n is the number of binary digits of the largest input number. Consequently, Euclid's algorithm runs in polynomial time.

Exercise 1.11. Show that we can compute the Jacobi symbol in polynomial time. *Hint.* Copy Euclid's algorithm.

Exercise 1.12. Prove: if p is an odd prime then $\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} \pmod{p}$ (Hint: use the fact that the multiplicative group mod p is cyclic).

Theorem 1.13. *Let N be an integer. Assume that N is not a prime and that N is not m^k for some $k \geq 2$. Then there exists $a \in \mathbb{Z}_N^*$ such that*

$$\left(\frac{a}{N}\right) \not\equiv a^{(N-1)/2} \pmod{N}. \tag{3}$$

Exercise 1.14. Show that if there exists $a \in Z_N^*$ satisfying (3) then at least half of the numbers in Z_N^* satisfy (3).

Exercise 1.15. Show that given an integer N we can check in polynomial time if N is of the form $N = m^k$ for some $k \geq 2$. (m, k are unknown integers.) *Hint.* Show that $k \leq \log_2 N$. Use binary search for each fixed k.

Theorem 1.13, and Exercises 1.11, 1.12, 1.14, 1.15 give us a polynomial-time randomized algorithm which on input N,

- if N is composite
  - with probability $\geq 1/2$ outputs a proof that N is composite,
  - otherwise outputs "don't know";
- if N is prime, always outputs "don't know."

Amplification. Note that by repeating this algorithm k times with independent coin flips, the probability that we never find a proof of compositeness when N is composite is reduced to $\leq 1/2^k$, so if we get "don't know" each time, it is a safe bet that N is prime. (How safe?)

2 Rabin-Miller primality testing algorithm

Exercise 2.1. Let p be a prime. Suppose that $a^{2t} \equiv 1 \pmod{p}$. Then $a^t \equiv \pm 1 \pmod{p}$.

The Miller-Rabin algorithm works as follows on input N. We assume N is a positive odd integer.

- It computes k such that 2^k is the largest power of 2 dividing $N - 1$.
- Then it picks random $1 \leq a \leq N - 1$. If $\gcd(a, n) \neq 1$ it outputs "composite."
- Otherwise it computes the sequence

$$a^{N-1} \bmod N, a^{(N-1)/2} \bmod N, \ldots, a^{(N-1)/2^k} \bmod N. \tag{4}$$

  If the sequence (4) does not start with 1 it outputs "composite." If the first element in the sequence which is not 1 is not -1 then it outputs "composite." Otherwise it outputs "don't know."

Exercise 2.2. Show that if N is an odd composite number then there exists $a \in \mathbb{Z}_N^*$ which causes the Miller-Rabin algorithm output "composite." Show that in fact at least half of $a \in \mathbb{Z}_N^*$ cause this output.

3 Lovász Toggle

Let $G = (V, E)$ be a graph with maximal degree Δ. Let r, b be positive integers such that $\Delta \leq r + b + 1$. Does there exist a coloring of the vertices red and blue such that

(i) every red vertex has at most r red neighbors; and

(ii) every blue vertex has at most b blue neighbors.

Theorem 3.1 (Lovász). *A coloring satisfying (i) and (ii) always exists.*

Lovász proved this theorem by showing that the following algorithm always terminates. We call a vertex *bad* (with respect to the current red/blue coloring) if it violates (i) or (ii).

Procedure "Lovász toggle"

Start from an arbitrary red/blue coloring of the vertices.

while a bad vertex exists,

pick a bad vertex and recolor it. **end** (**while**)

Exercise 3.2. Show that the algorithm terminates in $O(|E|)$ rounds, from any starting configuration. (A *round* is a cycle of the **while** loop, i. e., one vertex gets recolored in each round.)

Exercise$^+$ 3.3. Is it true that the algorithm terminates in $O(|V|)$ rounds? (The answer is not known to the instructor.)