Randomized Computation

We already have a poly-time deterministic algorithm for perfect matching.

Let's put together a Monte-Carlo algorithm for something we don't know how to do efficiently: *primality testing*.

Randomized Computation

Theorem: If N is prime, then for $1 \leq a < N$,

$$a^{N-1} = 1 \pmod{N}$$

This is a classical theorem of elementary number theory.

Randomized Computation

What if N is not prime?

Papadimitriou gives a table suggesting that if N is composite, then for at least half of $1 \leq a < N$,

$$a^{N-1} \not\equiv 1 \pmod{N}$$

If this conjecture is true, then here is a MC algorithm:

- pick a random $1 \leq a < N$,
- if $a^{N-1} \not\equiv N$ then *reject* (N is composite)
- otherwise *accept* (N is probably prime).

Does this algorithm work?

Randomized Computation

The algorithm does not work.

For all $1 \le a < 561$, $a^{560} = 1 \pmod{561}$.

But

$$561 = 3 \times 11 \times 17$$

- For all prime divisors p of $N = 561$, $(p-1)|(N-1)$.
- Called a *Carmichael number*.

Randomized Computation

Let's see if we can fix our primality testing algorithm

The equation

$$x^2 = a \pmod{p}$$

has either 2 roots, or none.

Lemma

- $a^{\frac{p-1}{2}} = \pm 1 \pmod{p}$
- if $a^{\frac{p-1}{2}} = 1 \pmod{p}$ then a has two square roots.
- if $a^{\frac{p-1}{2}} = -1 \pmod{p}$ then a has no square roots.

Randomized Computation

The expression $a^{(p-1)/2} \pmod p$

- indicates whether a has a square root modulo p
- abbreviated as $(a|p)$
- called *Legendre symbol*.

Randomized Computation

The law of Quadratic Reciprocity:

Theorem:

$$(p|q) \cdot (q|p) = (-1)^{\frac{p-1}{2}\frac{q-1}{2}}$$

In other words,

- $(p|q)$ and $(q|p)$ are the same,
- unless p and q are both $-1 \pmod 4$,
- in which case they are different.

Randomized Computation

A useful property:

$$
\begin{aligned}
(ab|p) &= (ab)^{(p-1)/2} \bmod p \\
&= a^{p-1}b^{(p-1)/2} \bmod p \\
&= (a|p)(b|p)
\end{aligned}
$$

And Another:

$$
\begin{aligned}
(a+p|p) &= (a+p)^{(p-1)/2} \bmod p \\
&= a^{(p-1)/2} \bmod p \\
&= (a|p)
\end{aligned}
$$

Randomized Computation

Let

$$N = q_1 \cdots q_n$$

where the q_i are odd primes, not necessarily distinct.

Define

$$(M|N) = \prod_{i=1}^{n} (M|q_i)$$

- No connection with square roots
- defined only when M, N relatively prime,
- and only when N is odd.

Randomized Computation

Useful properties remain:

$$
\begin{aligned}
(M_1 M_2 | N) &= (M_1|N)(M_2|N) \\
(M + N|N) &= (M|N) \\
(N|M) \cdot (M|N) &= (-1)^{\frac{M-1}{2}\frac{N-1}{2}}
\end{aligned}
$$

The last holds only if M and N are both odd.

Randomized Computation

Now we can calculate $(M|N)$ without factoring M or N.

Idea:

- like Euclidean GCD
- swap larger and smaller
- must beware of even numbers

Use the following fact

$$(2|N) = (-1)^{\frac{M^2-1}{8}}$$

Randomized Computation

Algorithm *legendre*(M, N).

1 If $M > N$ then

- return *legendre*$(M \bmod N, N)$

2 If $M = 2K$ then

- return $(-1)^{\frac{M^2-1}{8}} \cdot$ *legendre*(K, N)

3 Otherwise,

- return $(-1)^{\frac{M-1}{2}\frac{N-1}{2}} \cdot$ *legendre*(N, M)

Randomized Computation

1 If $M > N$ then

- return *legendre*$(M \bmod N, N)$

This step justified because

$$(M + N|N) = (M|N)$$

Randomized Computation

2 If $M = 2K$ then

- return $(-1)^{\frac{M^2-1}{8}} \cdot legendre(K, N)$

This step justified because

$$\begin{aligned} (2K|N) &= (2|N) \cdot (K|N) \\ (2|N) &= (-1)^{\frac{M^2-1}{8}} \end{aligned}$$

Algorithm *legendre*(M, N).

3 Otherwise

- return $(-1)^{\frac{M-1}{2}\frac{N-1}{2}} \cdot$ *legendre*(N, M)

This step justified because $N > M$ and M is odd.

Because $(M|N) = \pm 1$,

$$
\begin{aligned}
(N|M) \cdot (M|N) &= (-1)^{\frac{M-1}{2}\frac{N-1}{2}} \\
(N|M) &= (-1)^{\frac{M-1}{2}\frac{N-1}{2}} \cdot (M|N)
\end{aligned}
$$

Legendre Algorithm

Claim: The *legendre* algorithm takes time linear in the length of the binary representations of M and N

Each level of recursion halves one of the numbers.

Randomized Computation

Theorem: If N is an odd composite, then for at least *half* of the values $1 \ldots N-1$,

$$(M|N) \neq M^{\frac{N-1}{2}} \bmod N$$

Remark: By definition, if N is an odd prime, then for all values $1 \ldots N-1$,

$$(M|N) = M^{\frac{N-1}{2}} \bmod N$$

Eureka!

Randomized Computation

Algorithm *prime*(N)

- Generate a random $M \in 2 \ldots N-1$.
- If $gcd(M, N) \neq 1$, *reject*(N is composite).
- Compare $(M|N)$ and $M^{\frac{N-1}{2}} \bmod N$.
- If they disagree, *reject*(N is composite).
- If they agree, *accept*(N is probably prime).