Monte Carlo Algorithms[a]

- The randomized bipartite perfect matching algorithm is called a **Monte Carlo algorithm** in the sense that
  - If the algorithm finds that a matching exists, it is always correct (no **false positives**).
  - If the algorithm answers in the negative, then it may make an error (**false negatives**).

[a]Metropolis and Ulam (1949).

Monte Carlo Algorithms (concluded)

- The algorithm makes a false negative with probability ≤ 0.5.[a]
  - Note this probability refers to[b]

$$\text{prob}[\,\text{algorithm answers "no"} \mid G \text{ has a perfect matching}\,]$$

not

$$\text{prob}[\,G \text{ has a perfect matching} \mid \text{algorithm answers "no"}\,].$$

- This probability is *not* over the space of all graphs or determinants, but *over* the algorithm's own coin flips.
  - It holds for *any* bipartite graph.

[a] Equivalently, among the coin flip sequences, at most half of them lead to the wrong answer.

[b] In general, prob[algorithm answers "no" | input is a "yes" instance].

The Markov Inequality[a]

Lemma 61 *Let x be a random variable taking nonnegative integer values. Then for any $k > 0$,*

$$\text{prob}[\, x \geq kE[\, x\,]\,] \leq 1/k.$$

- Let p_i denote the probability that $x = i$.

$$\begin{aligned} E[\, x\,] &= \sum_i i p_i \\ &= \sum_{i < kE[\, x\,]} i p_i + \sum_{i \geq kE[\, x\,]} i p_i \\ &\geq kE[\, x\,] \times \text{prob}[x \geq kE[\, x\,]]. \end{aligned}$$

[a]Andrei Andreyevich Markov (1856–1922).

Andrei Andreyevich Markov (1856–1922)

An Application of Markov's Inequality

- Suppose algorithm C runs in expected time $T(n)$ and always gives the right answer.
- Consider an algorithm that runs C for time $kT(n)$ and rejects the input if C does not stop within the time bound.
- By Markov's inequality, this new algorithm runs in time $kT(n)$ and gives the wrong answer with probability $\leq 1/k$.

An Application of Markov's Inequality (concluded)

- By running this algorithm m times, we reduce the error probability to $\leq k^{-m}$.[a]
- Suppose, instead, we run the algorithm for the same running time $mkT(n)$ once and rejects the input if it does not stop within the time bound.
- By Markov's inequality, this new algorithm gives the wrong answer with probability $\leq 1/(mk)$.
- This is much worse than the previous algorithm's error probability of $\leq k^{-m}$ for the same amount of time.

[a] With the same input. Thanks to a question on December 7, 2010.

FSAT for k-SAT Formulas (p. 428)

- Let $\phi(x_1, x_2, \ldots, x_n)$ be a k-SAT formula.
- If ϕ is satisfiable, then return a satisfying truth assignment.
- Otherwise, return "no."
- We next propose a randomized algorithm for this problem.

A Random Walk Algorithm for ϕ in CNF Form

```
Start with an arbitrary truth assignment T;
for i = 1, 2, ..., r do
    if T ⊨ φ then
        return "φ is satisfiable with T";
    else
        Let c be an unsatisfied clause in φ under T; {All of
        its literals are false under T.}
        Pick any x of these literals at random;
        Modify T to make x true;
    end if
end for
return "φ is unsatisfiable";
```

3SAT vs. 2SAT Again

- Note that if ϕ is unsatisfiable, the algorithm will not refute it.
- The random walk algorithm needs expected exponential time for 3SAT.
  - In fact, it runs in expected $O((1.333\cdots+\epsilon)^n)$ time with $r = 3n$,[a] much better than $O(2^n)$.[b]
- We will show immediately that it works well for 2SAT.
- The state of the art as of 2006 is expected $O(1.322^n)$ time for 3SAT and expected $O(1.474^n)$ time for 4SAT.[c]

[a]Use this setting per run of the algorithm.
[b]Schöning (1999).
[c]Kwama and Tamaki (2004); Rolf (2006).

Random Walk Works for 2SAT[a]

Theorem 62 *Suppose the random walk algorithm with $r = 2n^2$ is applied to any satisfiable* 2SAT *problem with n variables. Then a satisfying truth assignment will be discovered with probability at least 0.5.*

- Let $\hat{T}$ be a truth assignment such that $\hat{T} \models \phi$.
- Assume our starting T differs from $\hat{T}$ in i values.
  - Their Hamming distance is i.
  - Recall T is arbitrary.
- Let $t(i)$ denote the expected number of repetitions of the flipping step until a satisfying truth assignment is found.

[a]Papadimitriou (1991).

The Proof

- It can be shown that $t(i)$ is finite.
- $t(0) = 0$ because it means that $T = \hat{T}$ and hence $T \models \phi$.
- If $T \neq \hat{T}$ or any other satisfying truth assignment, then we need to flip the coin at least once.
- We flip a coin to pick among the 2 literals of a clause not satisfied by the present T.
- At least one of the 2 literals is true under $\hat{T}$ because $\hat{T}$ satisfies all clauses.
- So we have at least 0.5 chance of moving closer to $\hat{T}$.

The Proof (continued)

- Thus
$$t(i) \leq \frac{t(i-1)+t(i+1)}{2}+1$$
for $0 < i < n$.
  - Inequality is used because, for example, T may differ from $\hat{T}$ in both literals.
- It must also hold that
$$t(n) \leq t(n-1)+1$$
because at $i = n$, we can only decrease i.

The Proof (continued)

- As we are only interested in upper bounds, we solve

$$\begin{aligned} x(0) &= 0 \\ x(n) &= x(n-1)+1 \\ x(i) &= \frac{x(i-1)+x(i+1)}{2}+1, \quad 0 < i < n \end{aligned}$$

- This is one-dimensional random walk with an absorbing barrier at $i=0$ and a reflecting barrier at $i=n$.

The Proof (continued)

- Add the equations up to obtain

$$\begin{aligned} & x(1) + x(2) + \cdots + x(n) \\ = \quad & \frac{x(0) + x(1) + 2x(2) + \cdots + 2x(n-2) + x(n-1) + x(n)}{2} \\ & +n + x(n-1). \end{aligned}$$

- Simplify to yield

$$\frac{x(1) + x(n) - x(n-1)}{2} = n.$$

- As $x(n) - x(n-1) = 1$, we have

$$x(1) = 2n - 1.$$

The Proof (continued)

- Iteratively, we obtain

$$\begin{aligned} x(2) &= 4n - 4, \\ &\vdots \\ x(i) &= 2in - i^2. \end{aligned}$$

- The worst case happens when $i = n$, in which case

$$x(n) = n^2.$$

The Proof (concluded)

- We therefore reach the conclusion that

$$t(i) \leq x(i) \leq x(n) = n^2.$$

- So the expected number of steps is at most n^2.
- The algorithm picks $r = 2n^2$.
  - This amounts to invoking the Markov inequality (p. 460) with $k = 2$, with the consequence of having a probability of 0.5.
- The proof does not yield a polynomial bound for 3SAT.[a]

[a]Contributed by Mr. Cheng-Yu Lee (R95922035) on November 8, 2006.

Boosting the Performance

- We can pick $r = 2mn^2$ to have an error probability of
$$\leq \frac{1}{2m}$$
by Markov's inequality.
- Alternatively, with the same running time, we can run the "$r = 2n^2$" algorithm m times.
- The error probability is now reduced to
$$\leq 2^{-m}.$$

Primality Tests

- PRIMES asks if a number N is a prime.
- The classic algorithm tests if $k \mid N$ for $k = 2, 3, \ldots, \sqrt{N}$.
- But it runs in $\Omega(2^{(\log_2 N)/2})$ steps.

Primality Tests (concluded)

- Suppose $N = PQ$ is a product of 2 distinct primes.
- The probability of success of the density attack (p. 409) is

$$\approx \frac{2}{\sqrt{N}}$$

  when $P \approx Q$.
- This probability is exponentially small in terms of the input length $\log_2 N$.

The Fermat Test for Primality

Fermat's "little" theorem on p. 412 suggests the following primality test for any given number N:

Pick a number a randomly from $\{1, 2, \ldots, N-1\}$;
if $a^{N-1} \neq 1 \bmod N$ **then**
    return "N is composite";
else
    return "N is a prime";
end if

The Fermat Test for Primality (concluded)

- Unfortunately, there are composite numbers called **Carmichael numbers** that will pass the Fermat test for *all* $a \in \{1, 2, \ldots, N-1\}$.[a]
  - The Fermat test will return "N is a prime" for all Carmichael numbers N.
- There are infinitely many Carmichael numbers.[b]
- In fact, the number of Carmichael numbers less than N exceeds $N^{2/7}$ for N large enough.

---

[a] Carmichael (1910).

[b] Alford, Granville, and Pomerance (1992).

Square Roots Modulo a Prime

- Equation $x^2 = a \bmod p$ has at most two (distinct) roots by Lemma 57 (p. 417).
  - The roots are called **square roots**.
  - Numbers a with square roots *and* $\gcd(a, p) = 1$ are called **quadratic residues**.
    - They are

$$1^2 \bmod p, 2^2 \bmod p, \ldots, (p-1)^2 \bmod p.$$

- We shall show that a number either has two roots or has none, and testing which one is true is trivial.[a]

[a] No efficient *deterministic* root-finding algorithms are known yet.

Euler's Test

Lemma 63 (Euler) *Let p be an odd prime and $a \neq 0 \bmod p$.*

1. *If*
$$a^{(p-1)/2} = 1 \bmod p,$$
then $x^2 = a \bmod p$ has two roots.

2. *If*
$$a^{(p-1)/2} \neq 1 \bmod p,$$
then
$$a^{(p-1)/2} = -1 \bmod p$$
and $x^2 = a \bmod p$ has no roots.

The Proof (continued)

- Let r be a primitive root of p.
- By Fermat's "little" theorem,

$$r^{(p-1)/2}$$

  is a square root of 1.
- So

$$r^{(p-1)/2} = 1 \text{ or } -1 \bmod p.$$

- But as r is a primitive root, $r^{(p-1)/2} \neq 1 \bmod p$.
- Hence

$$r^{(p-1)/2} = -1 \bmod p.$$

The Proof (continued)

- Let $a = r^k \bmod p$ for some k.
- Then

$$1 = a^{(p-1)/2} = r^{k(p-1)/2} = \left[r^{(p-1)/2} \right]^k = (-1)^k \bmod p.$$

- So k must be even.
- Suppose $a = r^{2j}$ for some $1 \leq j \leq (p-1)/2$.
- Then $a^{(p-1)/2} = r^{j(p-1)} = 1 \bmod p$, and a's two *distinct* roots are $r^j, -r^j (= r^{j+(p-1)/2} \bmod p)$.
  - If $r^j = -r^j \bmod p$, then $2r^j = 0 \bmod p$, which implies $r^j = 0 \bmod p$, a contradiction.

The Proof (continued)

- As $1 \leq j \leq (p-1)/2$, there are $(p-1)/2$ such a's.
- Each such a has 2 distinct square roots.
- The square roots of all the a's are distinct.
  - The square roots of different a's must be different.
- Hence the set of *square roots* is $\{1, 2, \ldots, p-1\}$.
- As a result, $a = r^{2j}$, $1 \leq j \leq (p-1)/2$, exhaust all the quadratic residues.

The Proof (concluded)

- If $a = r^{2j+1}$, then it has no roots because all the square roots have been taken.
- Now,

$$a^{(p-1)/2} = \left[r^{(p-1)/2} \right]^{2j+1} = (-1)^{2j+1} = -1 \bmod p.$$

The Legendre Symbol[a] and Quadratic Residuacity Test

- By Lemma 63 (p. 480) $a^{(p-1)/2} \bmod p = \pm 1$ for $a \neq 0 \bmod p$.
- For odd prime p, define the **Legendre symbol** $(a \,|\, p)$ as

$$
(a \,|\, p) = \begin{cases} 0 & \text{if } p \,|\, a, \\ 1 & \text{if } a \text{ is a quadratic residue modulo } p, \\ -1 & \text{if } a \text{ is a } \textbf{quadratic nonresidue} \text{ modulo } p. \end{cases}
$$

- Euler's test implies $a^{(p-1)/2} = (a \,|\, p) \bmod p$ for any odd prime p and any integer a.
- Note that $(ab|p) = (a|p)(b|p)$.

[a] Andrien-Marie Legendre (1752–1833).

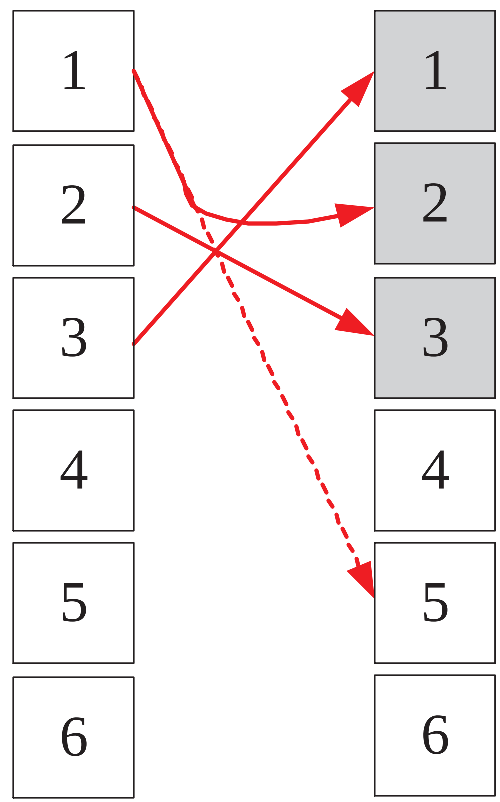
Gauss's Lemma

Lemma 64 (Gauss) *Let p and q be two odd primes. Then $(q|p) = (-1)^m$, where m is the number of residues in $R = \{ iq \bmod p : 1 \leq i \leq (p-1)/2 \}$ that are greater than $(p-1)/2$.*

- All residues in R are distinct.
  - If $iq = jq \bmod p$, then $p|(j-i)\, q$ or $p|q$.
- No two elements of R add up to p.
  - If $iq + jq = 0 \bmod p$, then $p|(i+j)$ or $p|q$.
  - But neither is possible.

The Proof (continued)

- Consider the set R' of residues that result from R if we replace each of the m elements $a \in R$ such that $a > (p-1)/2$ by $p - a$.
  - This is equivalent to performing $-a \bmod p$.
- All residues in R' are now at most $(p-1)/2$.
- In fact, $R' = \{1, 2, \ldots, (p-1)/2\}$ (see illustration next page).
  - Otherwise, two elements of R would add up to p, which has been shown to be impossible.



$p = 7$ and $q = 5$.

The Proof (concluded)

- Alternatively, $R' = \{\pm iq \bmod p : 1 \leq i \leq (p-1)/2\}$, where exactly m of the elements have the minus sign.
- Take the product of all elements in the two representations of R'.
- So

$$[(p-1)/2]! = (-1)^m q^{(p-1)/2} [(p-1)/2]! \bmod p.$$

- Because $\gcd([(p-1)/2]!, p) = 1$, the above implies

$$1 = (-1)^m q^{(p-1)/2} \bmod p.$$

Legendre's Law of Quadratic Reciprocity[a]

- Let p and q be two odd primes.
- The next result says their Legendre symbols are distinct if and only if both numbers are 3 mod 4.

Lemma 65 (Legendre (1785), Gauss)

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

[a]First stated by Euler in 1751. Legendre (1785) did not give a correct proof. Gauss proved the theorem when he was 19. He gave at least 6 different proofs during his life. The 152nd proof appeared in 1963.

The Proof (continued)

- Sum the elements of R' in the previous proof in mod2.
- On one hand, this is just $\sum_{i=1}^{(p-1)/2} i \bmod 2$.
- On the other hand, the sum equals

$$\begin{aligned} & \sum_{i=1}^{(p-1)/2} \left(qi - p \left\lfloor \frac{qi}{p} \right\rfloor \right) + mp \bmod 2 \\ = \quad & \left(q \sum_{i=1}^{(p-1)/2} i - p \sum_{i=1}^{(p-1)/2} \left\lfloor \frac{qi}{p} \right\rfloor \right) + mp \bmod 2. \end{aligned}$$

  – Signs are irrelevant under mod2.
  – m is as in Lemma 64 (p. 486).

The Proof (continued)

- Ignore odd multipliers to make the sum equal

$$\left(\sum_{i=1}^{(p-1)/2} i - \sum_{i=1}^{(p-1)/2} \left\lfloor \frac{qi}{p} \right\rfloor \right) + m \bmod 2.$$

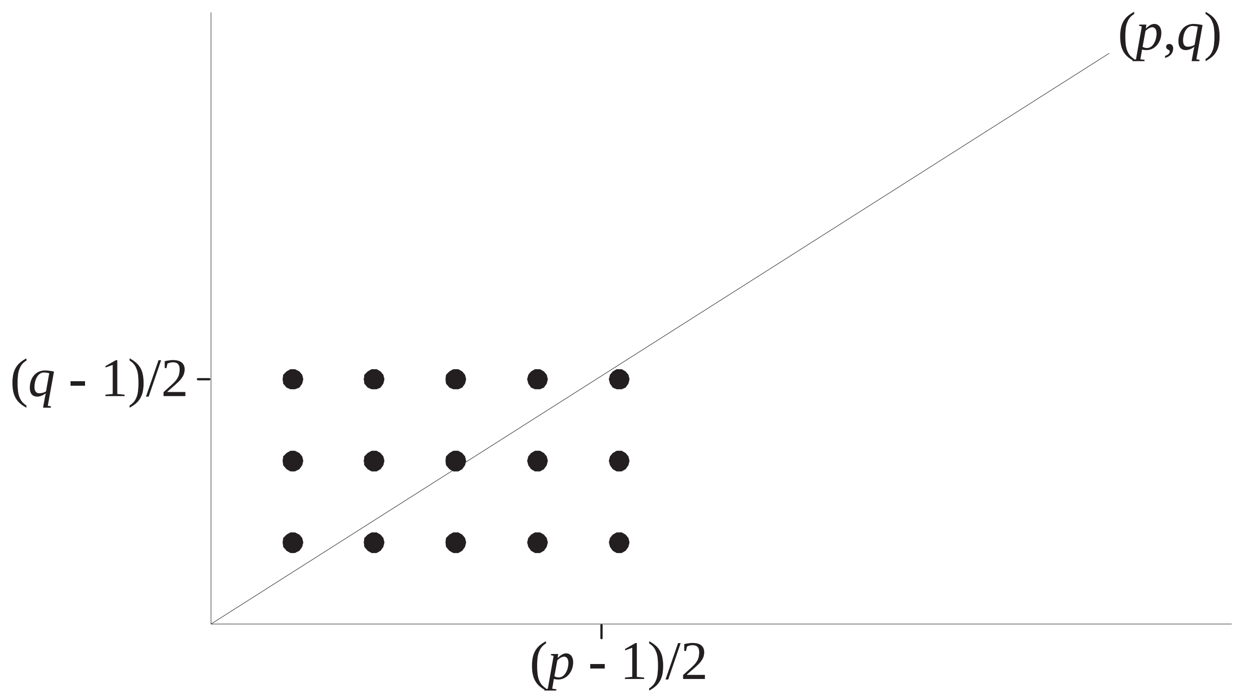
- Equate the above with $\sum_{i=1}^{(p-1)/2} i \bmod 2$ to obtain

$$m = \sum_{i=1}^{(p-1)/2} \left\lfloor \frac{qi}{p} \right\rfloor \bmod 2.$$

The Proof (concluded)

- $\sum_{i=1}^{(p-1)/2} \lfloor \frac{qi}{p} \rfloor$ is the number of integral points under the line
$$y = (q/p)\, x$$
for $1 \leq x \leq (p-1)/2$.
- Gauss's lemma (p. 486) says $(q|p) = (-1)^m$.
- Repeat the proof with p and q reversed.
- So $(p|q) = (-1)^{m'}$, where m' is the number of integral points *above* the line $y = (q/p)\, x$ for $1 \leq y \leq (q-1)/2$.
- As a result, $(p|q)(q|p) = (-1)^{m+m'}$.
- But $m + m'$ is the total number of integral points in the $\frac{p-1}{2} \times \frac{q-1}{2}$ rectangle, which is $\frac{p-1}{2} \frac{q-1}{2}$.

Eisenstein's Rectangle



$p = 11$ and $q = 7$.

The Jacobi Symbol[a]

- The Legendre symbol only works for odd *prime* moduli.
- The **Jacobi symbol** $(a \mid m)$ extends it to cases where m is not prime.
- Let $m = p_1 p_2 \cdots p_k$ be the prime factorization of m.
- When $m > 1$ is odd and $\gcd(a, m) = 1$, then

$$(a|m) = \prod_{i=1}^{k} (a \mid p_i).$$

  – Note that the Jacobi symbol equals ± 1.

  – It reduces to the Legendre symbol when m is a prime.

- Define $(a \mid 1) = 1$.

[a]Carl Jacobi (1804–1851).

Properties of the Jacobi Symbol

The Jacobi symbol has the following properties, for arguments for which it is defined.

1. $(ab \mid m) = (a \mid m)(b \mid m)$.
2. $(a \mid m_1 m_2) = (a \mid m_1)(a \mid m_2)$.
3. If $a = b \bmod m$, then $(a \mid m) = (b \mid m)$.
4. $(-1 \mid m) = (-1)^{(m-1)/2}$ (by Lemma 64 on p. 486).
5. $(2 \mid m) = (-1)^{(m^2-1)/8}$.[a]
6. If a and m are both odd, then $(a \mid m)(m \mid a) = (-1)^{(a-1)(m-1)/4}$.

[a] By Lemma 64 (p. 486) and some parity arguments.

Properties of the Jacobi Symbol (concluded)

- These properties allow us to calculate the Jacobi symbol without factorization.
- This situation is similar to the Euclidean algorithm.
- Note also that $(a \mid m) = 1/(a \mid m)$ because $(a \mid m) = \pm 1$.[a]

---

[a] Contributed by Mr. Huang, Kuan-Lin (B96902079, R00922018) on December 6, 2011.

Calculation of $(2200|999)$

$$
\begin{aligned}
(202|999) &= (2|999)(101|999) \\
&= (-1)^{(999^2-1)/8}(101|999) \\
&= (-1)^{124750}(101|999) = (101|999) \\
&= (-1)^{(100)(998)/4}(999|101) = (-1)^{24950}(999|101) \\
&= (999|101) = (90|101) = (-1)^{(101^2-1)/8}(45|101) \\
&= (-1)^{1275}(45|101) = -(45|101) \\
&= -(-1)^{(44)(100)/4}(101|45) = -(101|45) = -(11|45) \\
&= -(-1)^{(10)(44)/4}(45|11) = -(45|11) \\
&= -(1|11) = -1.
\end{aligned}
$$

A Result Generalizing Proposition 10.3 in the Textbook

Theorem 66 *The group of set* $\Phi(n)$ *under multiplication* mod n *has a primitive root if and only if* n *is either 1, 2, 4,* p^k*, or* $2p^k$ *for some nonnegative integer* k *and and odd prime* p*.*

This result is essential in the proof of the next lemma.

The Jacobi Symbol and Primality Test[a]

Lemma 67 *If* $(M|N) = M^{(N-1)/2} \bmod N$ *for all* $M \in \Phi(N)$*, then* N *is a prime. (Assume* N *is odd.)*

- Assume $N = mp$, where p is an odd prime, $\gcd(m, p) = 1$, and $m > 1$ (not necessarily prime).
- Let $r \in \Phi(p)$ such that $(r \,|\, p) = -1$.
- The Chinese remainder theorem says that there is an $M \in \Phi(N)$ such that

$$\begin{aligned} M &= r \bmod p, \\ M &= 1 \bmod m. \end{aligned}$$

[a]Mr. Clement Hsiao (`R88526067`) pointed out that the textbook's proof for Lemma 11.8 is incorrect while he was a senior in January 1999.

The Proof (continued)

- By the hypothesis,

$$M^{(N-1)/2} = (M \mid N) = (M \mid p)(M \mid m) = -1 \bmod N.$$

- Hence

$$M^{(N-1)/2} = -1 \bmod m.$$

- But because $M = 1 \bmod m$,

$$M^{(N-1)/2} = 1 \bmod m,$$

  a contradiction.

The Proof (continued)

- Second, assume that $N = p^a$, where p is an odd prime and $a \geq 2$.
- By Theorem 66 (p. 499), there exists a primitive root r modulo p^a.
- From the assumption,

$$M^{N-1} = \left[M^{(N-1)/2} \right]^2 = (M|N)^2 = 1 \bmod N$$

for all $M \in \Phi(N)$.

The Proof (continued)

- As $r \in \Phi(N)$ (prove it), we have
$$r^{N-1} = 1 \bmod N.$$
- As r's exponent modulo $N = p^a$ is $\phi(N) = p^{a-1}(p-1)$,
$$p^{a-1}(p-1) \,|\, N-1,$$
which implies that $p \,|\, N-1$.
- But this is impossible given that $p \,|\, N$.

The Proof (continued)

- Third, assume that $N = mp^a$, where p is an odd prime, $\gcd(m, p) = 1$, $m > 1$ (not necessarily prime), and a is even.
- The proof mimics that of the second case.
- By Theorem 66 (p. 499), there exists a primitive root r modulo p^a.
- From the assumption,

$$M^{N-1} = \left[M^{(N-1)/2} \right]^2 = (M|N)^2 = 1 \bmod N$$

for all $M \in \Phi(N)$.

The Proof (continued)

- In particular,
$$M^{N-1} = 1 \bmod p^a \tag{7}$$
for all $M \in \Phi(N)$.

- The Chinese remainder theorem says that there is an $M \in \Phi(N)$ such that
$$\begin{aligned} M &= r \bmod p^a, \\ M &= 1 \bmod m. \end{aligned}$$

- Because $M = r \bmod p^a$ and Eq. (7),
$$r^{N-1} = 1 \bmod p^a.$$

The Proof (concluded)

- As r's exponent modulo $N = p^a$ is $\phi(N) = p^{a-1}(p-1)$,
$$p^{a-1}(p-1) \mid N-1,$$
which implies that $p \mid N-1$.
- But this is impossible given that $p \mid N$.

The Number of Witnesses to Compositeness

Theorem 68 (Solovay and Strassen (1977)) *If N is an odd composite, then $(M|N) = M^{(N-1)/2} \bmod N$ for at most half of $M \in \Phi(N)$.*

- By Lemma 67 (p. 500) there is at least one $a \in \Phi(N)$ such that $(a|N) \neq a^{(N-1)/2} \bmod N$.
- Let $B = \{b_1, b_2, \ldots, b_k\} \subseteq \Phi(N)$ be the set of *all* distinct residues such that $(b_i|N) = b_i^{(N-1)/2} \bmod N$.
- Let $aB = \{ab_i \bmod N : i = 1, 2, \ldots, k\}$.

The Proof (concluded)

- $|aB| = k$.
  - $ab_i = ab_j \bmod N$ implies $N|a(b_i - b_j)$, which is impossible because $\gcd(a, N) = 1$ and $N > |b_i - b_j|$.
- $aB \cap B = \emptyset$ because

$$(ab_i)^{(N-1)/2} = a^{(N-1)/2} b_i^{(N-1)/2} \neq (a|N)(b_i|N) = (ab_i|N).$$

- Combining the above two results, we know

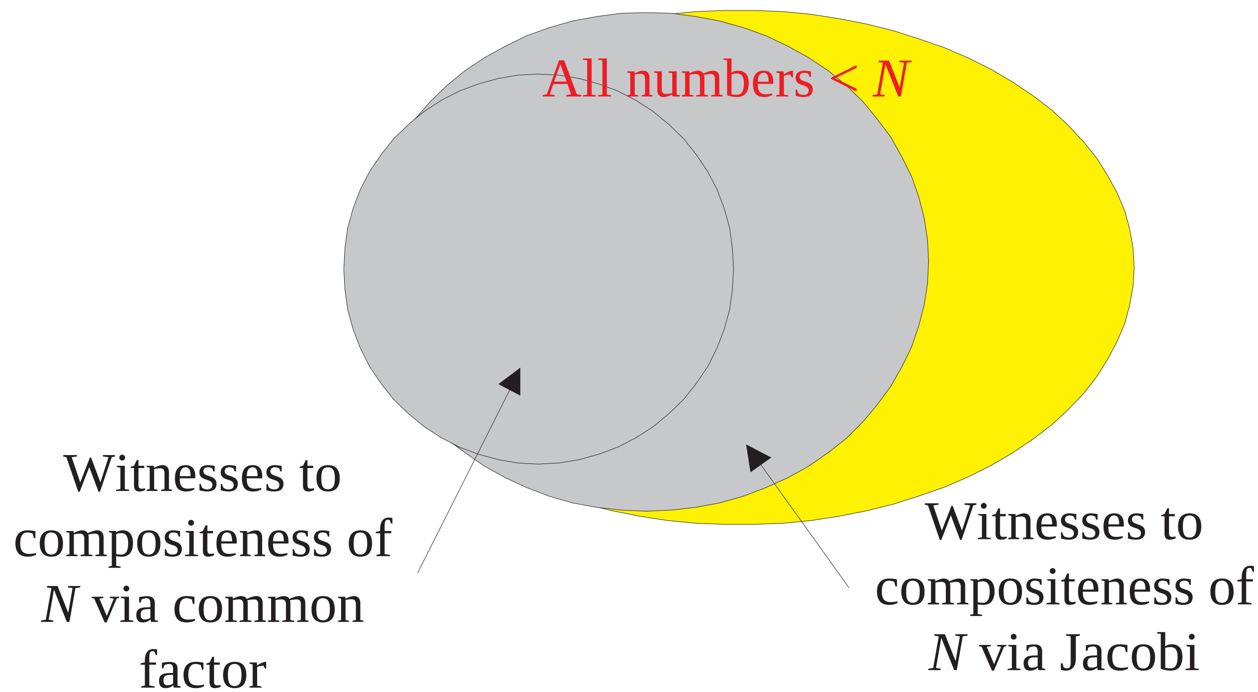
$$\frac{|\,B\,|}{\phi(N)} \leq \frac{|\,B\,|}{|\,B \cup aB\,|} = 0.5.$$

```
if N is even but N ≠ 2 then
    return "N is composite";
else if N = 2 then
    return "N is a prime";
end if
Pick M ∈ {2, 3, ..., N − 1} randomly;
if gcd(M, N) > 1 then
    return "N is composite";
else
    if (M|N) ≠ M^((N−1)/2) mod N then
        return "N is composite";
    else
        return "N is a prime";
    end if
end if
```

Analysis

- The algorithm certainly runs in polynomial time.
- There are no false positives (for COMPOSITENESS).
  - When the algorithm says the number is composite, it is always correct.
- The probability of a false negative is at most one half.
  - If the input is composite, then the probability that the algorithm says the number is a prime is ≤ 0.5.
- So it is a Monte Carlo algorithm for COMPOSITENESS.

The Improved Density Attack for COMPOSITENESS



Randomized Complexity Classes; RP

- Let N be a polynomial-time precise NTM that runs in time $p(n)$ and has 2 nondeterministic choices at each step.
- N is a **polynomial Monte Carlo Turing machine** for a language L if the following conditions hold:
  - If $x \in L$, then at least half of the $2^{p(n)}$ computation paths of N on x halt with "yes" where $n = |\, x \,|$.
  - If $x \notin L$, then all computation paths halt with "no."
- The class of all languages with polynomial Monte Carlo TMs is denoted **RP** (**randomized polynomial time**).[a]

[a]Adleman and Manders (1977).

Comments on RP

- Nondeterministic steps can be seen as fair coin flips.
- There are no false positive answers.
- The probability of false negatives, $1 - \epsilon$, is at most 0.5.
- But any constant between 0 and 1 can replace 0.5.
  - By repeating the algorithm $k = \lceil -\frac{1}{\log_2 1-\epsilon} \rceil$ times, the probability of false negatives becomes $(1 - \epsilon)^k \leq 0.5$.
- In fact, ϵ can be arbitrarily close to 0 as long as it is of the order $1/q(n)$ for some polynomial $q(n)$.
  - $-\frac{1}{\log_2 1-\epsilon} = O(\frac{1}{\epsilon}) = O(q(n))$.

Where RP Fits

- $P \subseteq RP \subseteq NP$.
  - A deterministic TM is like a Monte Carlo TM except that all the coin flips are ignored.
  - A Monte Carlo TM is an NTM with extra demands on the number of accepting paths.
- COMPOSITENESS $\in$ RP; PRIMES $\in$ coRP; PRIMES $\in$ RP.[a]
  - In fact, PRIMES $\in$ P.[b]
- $RP \cup coRP$ is an alternative "plausible" notion of efficient computation.

---

[a] Adleman and Huang (1987).
[b] Agrawal, Kayal, and Saxena (2002).

ZPP[a] (Zero Probabilistic Polynomial)

- The class **ZPP** is defined as $\mathrm{RP} \cap \mathrm{coRP}$.
- A language in ZPP has *two* Monte Carlo algorithms, one with no false positives and the other with no false negatives.
- If we repeatedly run both Monte Carlo algorithms, *eventually* one definite answer will come (unlike RP).
  - A *positive* answer from the one without false positives.
  - A *negative* answer from the one without false negatives.

[a] Gill (1977).

The ZPP Algorithm (**Las Vegas**)

{Suppose $L \in \mathrm{ZPP}$.}
{N_1 has no false positives, and N_2 has no false negatives.}
while `true` **do**
    if $N_1(x) =$ "yes" **then**
        return "yes";
    end if
    if $N_2(x) =$ "no" **then**
        return "no";
    end if
end while

ZPP (concluded)

- The *expected* running time for the correct answer to emerge is polynomial.
  - The probability that a run of the 2 algorithms does not generate a definite answer is 0.5 (why?).
  - Let $p(n)$ be the running time of each run of the while-loop.
  - The expected running time for a definite answer is

$$\sum_{i=1}^{\infty} 0.5^i i p(n) = 2p(n).$$

- Essentially, ZPP is the class of problems that can be solved without errors in expected polynomial time.