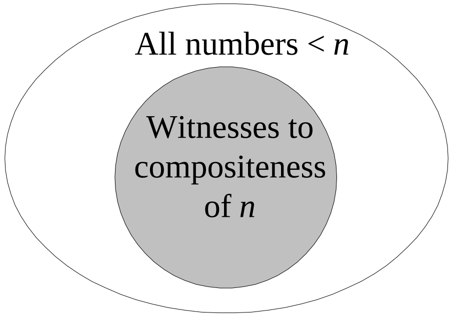
The Density Attack for PRIMES



- It works, but does it work well?

The Chinese Remainder Theorem

- Let $n = n_1 n_2 \cdots n_k$, where n_i are pairwise relatively prime.
- For any integers $a_1, a_2, \ldots, a_k$, the set of simultaneous equations

$$\begin{aligned} x &= a_1 \bmod n_1 \\ x &= a_2 \bmod n_2 \\ &\vdots \\ x &= a_k \bmod n_k \end{aligned}$$

has a unique solution modulo n for the unknown x.

Fermat's "Little" Theorem[a]

Lemma 56 *For all* $0 < a < p$, $a^{p-1} = 1 \bmod p$.

- Consider $a\Phi(p) = \{am \bmod p : m \in \Phi(p)\}$.
- $a\Phi(p) = \Phi(p)$.
  - Suppose $am = am' \bmod p$ for $m > m'$, where $m, m' \in \Phi(p)$.
  - That means $a(m - m') = 0 \bmod p$, and p divides a or $m - m'$, which is impossible.
- Hence $(p-1)! = a^{p-1}(p-1)! \bmod p$.
- Finally, $(a^{p-1} - 1) = 0 \bmod p$ because $p \not| (p-1)!$.

[a] Pierre de Fermat (1601–1665).

The Fermat-Euler Theorem

Corollary 57 *For all $a \in \Phi(n)$, $a^{\phi(n)} = 1 \bmod n$.*

- As $12 = 2^2 \times 3$,

$$\phi(12) = 12 \times \left(1 - \frac{1}{2}\right)\left(1 - \frac{1}{3}\right) = 4$$

- In fact, $\Phi(12) = \{1, 5, 7, 11\}$.
- For example,

$$5^4 = 625 = 1 \bmod 12.$$

Exponents

- The **exponent** of $m \in \Phi(p)$ is the least $k \in \mathbb{Z}^+$ such that

$$m^k = 1 \bmod p.$$

- Every residue $s \in \Phi(p)$ has an exponent.
  - $1, s, s^2, s^3, \ldots$ eventually repeats itself, say $s^i = s^j \bmod p$, which means $s^{j-i} = 1 \bmod p$.
- If the exponent of m is k and $m^\ell = 1 \bmod p$, then $k | \ell$.
  - Otherwise, $\ell = qk + a$ for $0 < a < k$, and $m^\ell = m^{qk+a} = m^a = 1 \bmod p$, a contradiction.

Lemma 58 *Any nonzero polynomial of degree k has at most k distinct roots modulo p.*

Exponents and Primitive Roots

- From Fermat's "little" theorem, all exponents divide $p - 1$.
- A primitive root of p is thus a number with exponent $p - 1$.
- Let $R(k)$ denote the total number of residues in $\Phi(p)$ that have exponent k.
- We already knew that $R(k) = 0$ for $k \not| (p - 1)$.
- Any $a \in \Phi(p)$ of exponent k satisfies $x^k = 1 \bmod p$.
- Hence there are at most k residues of exponent k, i.e., $R(k) \leq k$.

Size of $R(k)$

- Let s be a residue of exponent k.
- $1, s, s^2, \ldots, s^{k-1}$ are all distinct modulo p.
  - Otherwise, $s^i = s^j \bmod p$ with $i < j$ and s is of exponent $j - i < k$, a contradiction.
- As all these k distinct numbers satisfy $x^k = 1 \bmod p$, they are all the solutions of $x^k = 1 \bmod p$.
- But do all of them have exponent k (i.e., $R(k) = k$)?
- And if not (i.e., $R(k) < k$), how many of them do?

Size of $R(k)$ (continued)

- Suppose $\ell < k$ and $\ell \notin \Phi(k)$ with $\gcd(\ell, k) = d > 1$.
- Then
$$(s^\ell)^{k/d} = 1 \bmod p.$$
- Therefore, s^ℓ has exponent at most k/d, which is less than k.
- We conclude that
$$R(k) \leq \phi(k).$$

Size of $R(k)$ (concluded)

- Because all $p-1$ residues have an exponent,
$$p - 1 = \sum_{k|(p-1)} R(k) \leq \sum_{k|(p-1)} \phi(k) = p - 1$$
by Lemma 54 on p. 331.
- Hence
$$R(k) = \begin{cases} \phi(k) & \text{when } k|(p-1) \\ 0 & \text{otherwise} \end{cases}$$
- In particular, $R(p-1) = \phi(p-1) > 0$, and p has at least one primitive root.
- This proves one direction of Theorem 50 (p. 324).

A Few Calculations

- Let $p = 13$.
- From p. 338, we know $\phi(p-1) = 4$.
- Hence $R(12) = 4$.
- And there are 4 primitives roots of p.
- As $\Phi(p-1) = \{1, 5, 7, 11\}$, the primitive roots are g^1, g^5, g^7, g^{11} for any primitive root g.

The Other Direction of Theorem 50 (p. 324)

- Suppose p is not a prime.
- We proceed to show that no primitive roots exist.
- Suppose $r^{p-1} = 1 \bmod p$, the 1st condition of the primitive root on p. 324.
- We will show that the 2nd condition must be violated.
- $r^{\phi(p)} = 1 \bmod p$ by the Fermat-Euler theorem (p. 338).
- Because p is not a prime, $\phi(p) < p - 1$.

The Other Direction of Theorem 50 (concluded)

- Let k be the smallest integer such that $r^k = 1 \bmod p$.
- As $k | \phi(p)$, $k < p - 1$.
- Let q be a prime divisor of $(p-1)/k > 1$.
- Then $k | (p-1)/q$.
- Therefore, by virtue of the definition of k,

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

- But this violates the 2nd condition of the primitive root on p. 324.

Randomized Algorithms[a]

- Randomized algorithms flip unbiased coins.
- There are important problems for which there are no known efficient *deterministic* algorithms but for which very efficient randomized algorithms exist.
  - Primality tests, extraction of square roots, etc.
- There are problems where randomization is *necessary*.
  - Secure protocols.
- Are randomized algorithms algorithms[b]?

[a]Rabin, 1976, Solovay and Strassen, 1977.

[b]"Truth is so delicate that one has only to depart the least bit from it to fall into error." — *The Provincial Letters*, Pascal (1623–1662).

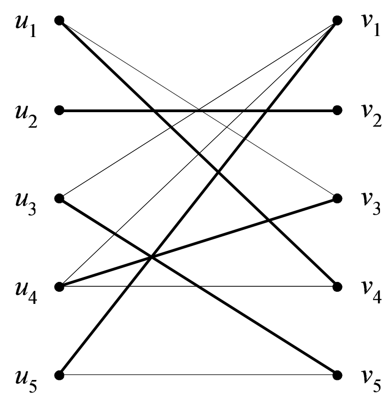
Bipartite Perfect Matching

- We are given a **bipartite graph** $G = (U, V, E)$.
  - $U = \{u_1, u_2, \ldots, u_n\}$.
  - $V = \{v_1, v_2, \ldots, v_n\}$.
  - $E \subseteq U \times V$.
- We are asked if there is a **perfect matching**.
  - A permutation π of $\{1, 2, \ldots, n\}$ such that

$$(u_i, v_{\pi(i)}) \in E$$

for all $u_i \in U$.

A Perfect Matching



Symbolic Determinants

- Given a bipartite graph G, construct the $n \times n$ matrix A^G whose (i, j)th entry A^G_{ij} is a variable x_{ij} if $(u_i, v_j) \in E$ and zero otherwise.
- The **determinant** of A^G is

$$\det(A^G) = \sum_{\pi} \sigma(\pi) \prod_{i=1}^{n} A^G_{i,\pi(i)}, \tag{5}$$

where π ranges over all permutations of n elements and $\sigma(\pi)$ is 1 if π is the product of an even number of transpositions and -1 otherwise.

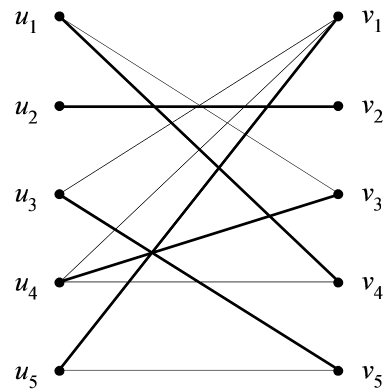
Determinant and Bipartite Perfect Matching

In $\sum_{\pi} \sigma(\pi) \prod_{i=1}^{n} A^G_{i,\pi(i)}$, note the following:

- Each summand corresponds to a possible prefect matching π.
- As all variables appear only *once*, all of these summands are different monomials and will not cancel.

Proposition 59 (Edmonds, 1967) *G has a perfect matching if and only if* $\det(A^G)$ *is not identically zero.*

A Perfect Matching in a Bipartite Graph



The Perfect Matching in the Determinant

- The matrix is

$$A^G = \begin{bmatrix} 0 & 0 & x_{13} & \boxed{x_{14}} & 0 \\ 0 & \boxed{x_{22}} & 0 & 0 & 0 \\ x_{31} & 0 & 0 & 0 & \boxed{x_{35}} \\ x_{41} & 0 & \boxed{x_{43}} & x_{44} & 0 \\ \boxed{x_{51}} & 0 & 0 & 0 & x_{55} \end{bmatrix}.$$

- $\det(A^G)$ contains term $x_{14}x_{22}x_{35}x_{43}x_{51}$, which denotes a perfect matching.

How To Test If a Polynomial Is Identically Zero?

- $\det(A^G)$ is a polynomial in n^2 variables.
- There are exponentially many terms in $\det(A^G)$.
- Expanding the determinant polynomial is not feasible.
  - Too many terms.
- Observation: If $\det(A^G)$ is *identically zero*, then it remains zero if we substitute *arbitrary* integers for the variables $x_{11}, \ldots, x_{nn}$.
- What is the likelihood of obtaining a zero when $\det(A^G)$ is *not* identically zero?

Number of Roots of a Polynomials

Lemma 60 (Schwartz, 1980) *Let $p(x_1, x_2, \ldots, x_m) \not\equiv 0$ be a polynomial in m variables each of degree at most d. Let $M \in \mathbb{Z}^+$. Then the number of m-tuples*

$$(x_1, x_2, \ldots, x_m) \in \{0, 1, \ldots, M-1\}^m$$

such that $p(x_1, x_2, \ldots, x_m) = 0$ is

$$\leq mdM^{m-1}.$$

- By induction on m.

Density Attack

- The density of roots in the domain is at most

$$\frac{mdM^{m-1}}{M^m} = \frac{md}{M}.$$

- This suggests a sampling algorithm.

A Randomized Bipartite Perfect Matching Algorithm[a]

Choose n^2 integers $i_{11}, \ldots, i_{nn}$ from $\{0, 1, \ldots, b-1\}$ randomly;
Calculate $\det(A^G(i_{11}, \ldots, i_{nn}))$ by Gaussian elimination;
if $\det(A^G(i_{11}, \ldots, i_{nn})) \neq 0$ **then**
    return "G has a perfect matching";
else
    return "G has no perfect matchings";
end if

[a]Lovász, 1979.

Analysis

- Pick b such that $b^{n^2} = 2n^2$.
- If G has no perfect matchings, the algorithm will always be correct.
- Suppose G has a perfect matching.
  - The algorithm will answer incorrectly with probability at most $n^2 d/b = 0.5$ because $d = 1$.
  - Repeat the algorithm *independently* k times and output "G has no perfect matchings" if all of the k runs say so.
  - The error probability is now reduced to at most 2^{-k}.

Monte Carlo Algorithms

- 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 negative**).
- The probability that the algorithm makes a false negative is at most 0.5.
- 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.

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).

An Application of Markov's Inequality

- 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 correct answer with probability at least $1 - (1/k)$.
- By running this algorithm m times, we reduce the error probability to $\leq k^{-m}$.

A Random Walk Algorithm for ϕ in CNF Form

Start with an *arbitrary* truth assignment T;
for $i = 1, 2, \ldots, r$ **do**
  if $T \models \phi$ **then**
    return "ϕ is satisfiable";
  else
    Let c be an unsatisfiable 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 and 2SAT Again

- Note that if ϕ is unsatisfiable, the algorithm will not refute it.
- The random walk algorithm runs in exponential time for 3SAT.
- But we will show that it works well for 2SAT.

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.*

The Proof

- Let $\hat{T}$ be a truth assignment such that $\hat{T} \models \phi$.
- Let $t(i)$ denote the expected number of repetitions of the flipping step until a satisfying truth assignment is found if our starting T differs from $\hat{T}$ in i values.
  - Their Hamming distance is i.
- 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 T is not equal to any other satisfying truth assignment, then we need to flip at least once.

The Proof (continued)

- We flip 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 $\hat{T}$.
- 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.

The Proof (continued)

- It must also hold that

$$t(n) \leq t(n-1) + 1$$

because at $i = n$, we can only decrease i.

- 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 a reflecting and an absorbing barrier.

The Proof (continued)

- Add the equations up to obtain

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

- 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 a running time $2n^2$.
- This amounts to invoking the Markov inequality (p. 360) with $k = 2$, with the consequence of having a probability of 0.5.

Boosting the Performance

- We can pick $r = 2mn^2$ to have an error probability of $\leq (2m)^{-1}$ by Markov's inequality.
- Alternatively, with the same running time, we can run the "$r = 2n^2$" algorithm m times.
- But the error probability is reduced to $\leq 2^{-m}$!
- The gain comes from the fact that Markov's inequality does not take advantage of any specific feature of the random variable.
- The gain also comes from the fact that the two algorithms are different.

Primality Tests

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

The Density Attack for PRIMES

```
Pick k ∈ {2, ..., p − 1} randomly; {Assume p > 2.}
if k | p then
  return "N is a composite";
else
  return "N is a prime";
end if
```

The probability of success when p is composite is $1 - \phi(p)/p$.

The Fermat Test for Primality

- Fermat's "little" theorem on p. 337 suggests the following primality test for any given number p:
  - Pick a number a randomly from $\{1, 2, \ldots, p-1\}$.
  - If $a^{p-1} \neq 1 \bmod p$, then declare "p is composite."
  - Otherwise, declare "p is probably prime."
- Unfortunately, there are composite numbers called **Carmichael numbers** that will pass the Fermat test for *all* $a \in \{1, 2, \ldots, p-1\}$.
- It is only recently that Carmichael numbers are known to be infinite in number.

Square Roots Modulo a Prime

- Equation $x^2 = a \bmod p$ has at most two (distinct) roots by Lemma 58 on p. 339.
  - The roots are called **square roots**.
  - Numbers a with square roots and $\gcd(a, p) = 1$ are called **quadratic residues**:
    $1^2 \bmod p, 2^2 \bmod p, \ldots, (p-1)^2 \bmod p$.
- We shall show that a number either has two roots or has none, and testing which is true is trivial.
- We remark that there are no known efficient *deterministic* algorithms to find the roots.

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.*

- Let r be a primitive root of p.
- If $a = r^{2j}$, then $a^{(p-1)/2} = r^{j(p-1)} = 1 \bmod p$ and its two distinct roots are $r^j, -r^j (= r^{j+(p-1)/2})$.

The Proof (concluded)

- Since there are $(p-1)/2$ such a's, and each such a has two distinct roots, we have run out of *square roots*.
  - $\{c : c^2 = a \bmod p\} = \{1, 2, \ldots, p-1\}$.
- If $a = r^{2j+1}$, then it has no roots because all the square roots are taken.
- By Fermat's "little" theorem, $r^{(p-1)/2}$ is a square root of 1, so $r^{(p-1)/2} = \pm 1 \bmod p$.
- But as r is a primitive root, $r^{(p-1)/2} = -1 \bmod p$.
- $a^{(p-1)/2} = (r^{(p-1)/2})^{2j+1} = (-1)^{2j+1} = -1 \bmod p$.