Fingerprinting

Fast verification by random evaluation

The first topic in the randomization module

The opening lecture used graph cuts to preview the course's three strands. This lecture begins the randomized-algorithms strand in earnest. Its recurring move is to replace a large object by a small random image: equal objects always have equal images, while unequal objects collide only with controlled probability. The same proof pattern verifies a matrix product, tests a polynomial identity, compresses communication, detects a perfect matching, searches a text, and checks a data stream.

Advanced Algorithms teaching team

Nanjing University

Learning goals and reading routes

After this lecture, you should be able to

1. prove Freivalds' one-sided error bound and amplify it;
2. distinguish a formal polynomial from the function it induces, and prove the Schwartz–Zippel theorem over an arbitrary field;
3. design and analyse a short fingerprint for Equality;
4. recognize determinant, rolling-hash, and product-polynomial fingerprints; and
5. state every runtime and error guarantee together with the field size, randomness, and computational model that make it valid.

For a first pass, read sections 1 and 2, the statement and proof of theorem 3.3, the polynomial protocol in section 4, and one of sections 5 to 7. The random-prime variants are a second view of the same collision principle. The appendices contain only the probability, prime-counting, finite-field, and polynomial tools used by the algorithms; no algorithm is deferred there.

Unless stated otherwise, arithmetic complexity counts field operations. For bit and streaming bounds we say explicitly how large one field element is. All logarithms in communication and space bounds are base two; changing the base affects only constants. For a positive integer n, write $[n] = \{1, \dots, n\}$.

Contents

1 The fingerprinting idea

Suppose direct comparison of objects from a large universe is expensive. A *fingerprint family* is a randomized collection of maps $h_s : \mathcal{U} \to \mathcal{R}$, indexed by a short random seed s. We seek three properties:

(i) **perfect completeness:** $X = Y$ implies $h_s(X) = h_s(Y)$ for every seed s;

(ii) **soundness:** for $X \neq Y$, $\mathbb{P}_s[h_s(X) = h_s(Y)] \leq \varepsilon$; and

(iii) **economy:** the seed, fingerprint, and evaluation procedure are much smaller than the original representation.

The range cannot usually be both small and collision-free. Randomization does not eliminate collisions; it prevents a fixed unequal pair from knowing *which* collision to exploit. Throughout this note, the input is fixed before the seed is sampled. If an adversary may choose an object after seeing the seed, a fresh or hidden seed is needed and the collision proof must be revisited.

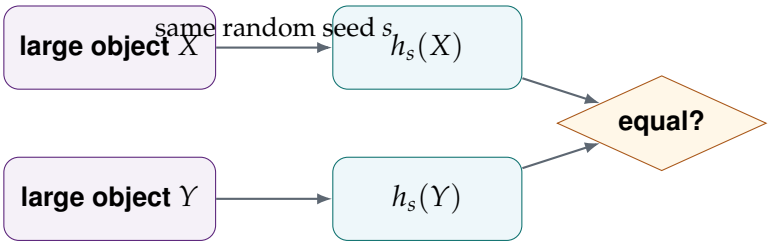


Figure 1: Fingerprinting replaces a large comparison by two evaluations and a small comparison. The analysis is always an analysis of collisions.

Objects or claim	Random image	Why a false equality is rare
Matrices AB **and** C	$(AB)r$ and Cr	a nonzero row gives one linear constraint
Polynomials f **and** 0	$f(r_1, \ldots, r_m)$	a nonzero degree-d polynomial has few roots
Bit strings x, y	polynomial values or residues	the difference has few roots or prime divisors
Multisets A, B	$\prod_{a \in A}(r - a)$	unequal root multiplicities give unequal polynomials

Table 1: The representation changes, but each proof identifies a nonzero witness and bounds the chance that random evaluation hides it.

One-sided error has a direction

In an equality test, equal objects are never rejected and unequal objects may collide: the only error is a false claim of equality. In the perfect-matching test later, a graph with no perfect matching is never accepted, while a graph that has one may evaluate to zero and be rejected. Always name the direction of the error instead of saying only "correct with high probability."

2 Freivalds' verifier for matrix multiplication

Given $A, B, C \in \mathbb{F}^{n \times n}$, we want to verify the assertion $AB = C$. Recomputing the product classically takes $O(n^3)$ field operations. Freivalds' algorithm instead asks whether the two linear maps agree on one random vector [3].

Freivalds(A, B, C)

1. Choose $r \in \{0,1\}^n$ uniformly at random.
2. Compute $u = Br$, $v = Au$, and $w = Cr$.
3. Accept if $v = w$; otherwise reject.

The parenthesization matters: the verifier performs three dense matrix–vector products, not one matrix–matrix product. It therefore uses $O(n^2)$ field operations, n random bits, and $O(n)$ auxiliary field elements. Reading three dense matrices already costs $\Theta(n^2)$ words.

Theorem 2.1 (Freivalds' guarantee)**.** *For matrices over any field,*

$$AB = C \quad \Longrightarrow \quad \mathbb{P}[\mathit{accept}] = 1,$$

whereas

$$AB \neq C \quad \Longrightarrow \quad \mathbb{P}[\mathit{accept}] \leq \frac{1}{2}.$$

Proof. If $AB = C$, associativity gives $A(Br) = Cr$ for every r.

Now suppose $M = AB - C \neq 0$. Choose a nonzero row i of M, and choose j with $M_{ij} \neq 0$. Condition on all coordinates of r except r_j. The equation $(Mr)_i = 0$ becomes

$$M_{ij} r_j = -\sum_{k \neq j} M_{ik} r_k.$$

Because M_{ij} is invertible, at most one field value of r_j satisfies this equation. The two candidates 0 and 1 are distinct in every field, so a uniform $r_j \in \{0,1\}$ satisfies it with probability at most $1/2$. Acceptance requires the stronger event $Mr = 0$, hence its probability is also at most $1/2$. □

Corollary 2.2 (Independent amplification)**.** *Run the verifier independently t times and accept only if every run accepts. The work is $O(tn^2)$ field operations; a wrong product is accepted with probability at most 2^{-t}. Thus $t = \lceil \log_2(1/\delta) \rceil$ gives error at most δ.*

What the proof really used

The set $\{0,1\}$ is not special. If every coordinate is uniform in a finite set $S \subseteq \mathbb{F}$, the same conditioning argument gives error at most $1/|S|$. The proof exposes the template used throughout the lecture: find a nonzero coordinate or coefficient, condition on everything else, and show that only a few random choices can hide it.

Arithmetic is exact in this theorem

Over floating-point arithmetic, roundoff makes the predicate $A(Br) = Cr$ model-dependent. The theorem is about exact arithmetic in a field, such as $\mathbb{F}_p$ or the rationals under a unit-cost abstraction. Bit complexity over the rationals must also account for growing numerator and denominator sizes.

3 Polynomial identity testing

Freivalds' verifier tests a linear identity. Polynomial identity testing (PIT) asks the same question for an arbitrary polynomial:

$$\text{given } f \in \mathbb{F}[x_1, \ldots, x_m], \qquad \text{is } f \equiv 0?$$

The symbol $\equiv 0$ means that every formal coefficient is zero. This is not always the same as saying $f(a) = 0$ for every $a \in \mathbb{F}^m$; over $\mathbb{F}_p$, for example, the nonzero polynomial $x^p - x$ vanishes at every field element. The field size and degree therefore belong in every black-box PIT statement.

We assume that f can be evaluated efficiently—perhaps by an arithmetic circuit, a determinant, or a product formula—even when expanding it into monomials would be prohibitive. Its *total degree* is the largest sum of exponents in a monomial with nonzero coefficient.

Example 3.1 (A polynomial with compact representations)**.** For the Vandermonde matrix $V_{ij} = x_i^{j-1}$,

$$\det V = \prod_{1 \le j < i \le m} (x_i - x_j).$$

The determinant and product both give short evaluation procedures, whereas a full determinant expansion ranges over $m!$ permutations. Black-box PIT charges for evaluating the chosen representation; it never requires the expanded coefficient list.

3.1 The univariate root bound

Lemma 3.2 (Root bound)**.** *A nonzero polynomial $f \in \mathbb{F}[x]$ of degree d has at most d distinct roots in $\mathbb{F}$.*

Proof. Induct on d. The claim is immediate for $d = 0$. If a is a root of a positive-degree polynomial, the factor theorem gives $f(x) = (x - a)g(x)$ with $\deg g = d - 1$. Every other root of f is a root of g, so there are at most $1 + (d - 1) = d$ roots. □

This is an algebraic root bound valid over every field; it is not the Fundamental Theorem of Algebra, which is a statement about factorization over $\mathbb{C}$. Consequently, if r is uniform in a finite $S \subseteq \mathbb{F}$,

$$f \not\equiv 0 \quad \Longrightarrow \quad \mathbb{P}[f(r) = 0] \le \frac{d}{|S|}.$$

The bound is useful only when $|S|$ is appreciably larger than d. It also gives a deterministic baseline: if the field contains $d + 1$ distinct points, a degree-at-most-d polynomial is identically zero exactly when it vanishes at all of them. Randomization replaces these $d + 1$ evaluations by one evaluation from a larger set and accepts a controlled chance of error.

3.2 Schwartz–Zippel

The multivariate extension was developed independently in work of DeMillo–Lipton, Schwartz, and Zippel [1, 6, 8].

Theorem 3.3 (Schwartz–Zippel)**.** *Let $f \in \mathbb{F}[x_1, \dots, x_m]$ be nonzero of total degree at most d. Let $S \subseteq \mathbb{F}$ be finite, and choose $r_1, \dots, r_m$ independently and uniformly from S. Then*

$$\mathbb{P}[f(r_1, \dots, r_m) = 0] \le \frac{d}{|S|}.$$

Equivalently, f has at most $d|S|^{m-1}$ zeros in the grid S^m.

Proof. We induct on m. The case $m = 1$ is lemma 3.2. For $m > 1$, view f as a polynomial in its last variable:

$$f(x_1, \dots, x_m) = \sum_{i=0}^{k} x_m^i f_i(x_1, \dots, x_{m-1}),$$

where $f_k \not\equiv 0$. Since f has total degree at most d, $\deg f_k \leq d-k$. Let E be the event $f_k(r_1,\dots,r_{m-1}) = 0$. By the induction hypothesis,

$$\mathbb{P}(E) \leq \frac{d-k}{|S|}.$$

Conditioned on E^c, substituting the first $m-1$ coordinates leaves a nonzero univariate polynomial in x_m of degree exactly k. The root bound therefore gives

$$\mathbb{P}[f(r_1,\dots,r_m) = 0 \mid E^c] \leq \frac{k}{|S|}.$$

Without needing to analyse the conditional probability on E, we obtain

$$\begin{aligned}\mathbb{P}[f(r) = 0] &\leq \mathbb{P}(E) + \mathbb{P}[f(r) = 0 \mid E^c]\mathbb{P}(E^c) \\ &\leq \frac{d-k}{|S|} + \frac{k}{|S|} = \frac{d}{|S|}.\end{aligned}$$

□

Black-box PIT by random evaluation

Given an evaluation procedure for a degree-d polynomial $f \in \mathbb{F}[x_1,\dots,x_m]$ and a finite $S \subseteq \mathbb{F}$:

1. choose $r_1,\dots,r_m$ independently and uniformly from S;
2. accept "identically zero" if $f(r_1,\dots,r_m) = 0$, and reject otherwise.

If $f \equiv 0$, the algorithm always accepts. If $f \not\equiv 0$, it falsely accepts with probability at most $d/|S|$.

Three hypotheses that cannot be dropped

Schwartz–Zippel concerns a *nonzero formal polynomial*, uses its *total degree*, and samples coordinates *independently*. If the base field is too small, pass to a suitable extension field or use a different encoding; merely evaluating at every point of a small field may reveal nothing.

4 Equality with logarithmic communication

Alice receives $x \in \{0,1\}^n$ and Bob receives $y \in \{0,1\}^n$. They want to compute

$$\mathrm{EQ}(x,y) = \begin{cases} 1, & x = y, \\ 0, & x \neq y, \end{cases}$$

while minimizing the number of communicated bits; local computation is free. This model, introduced by Yao [7], isolates information transfer from running time.

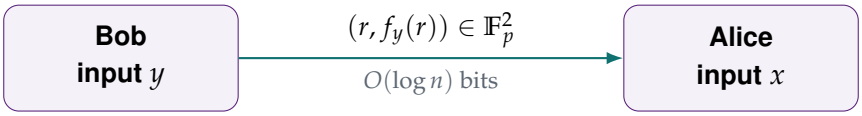
In a deterministic one-way protocol from Bob to Alice, fewer than n bits cannot suffice: fewer than 2^n messages force two distinct strings y, y' to produce the same message. Alice must accept that message on input $x = y$, and would then also accept it on the unequal pair (y, y'). Sending the whole string uses n bits, so the deterministic one-way cost is exactly n.

Interaction does not improve the exact deterministic bound. Every complete transcript of a deterministic protocol describes a rectangle $A \times B$ of input pairs. An accepting rectangle cannot contain both (x, x) and (y, y) for $x \neq y$, because it would then also contain the unequal pair (x, y). Thus the 2^n diagonal inputs require at least 2^n accepting transcripts, and a protocol of worst-case cost c has at most 2^c transcripts. Hence $c \geq n$ when one designated player produces the answer.

4.1 A polynomial fingerprint

Fix once and for all a prime p with $n^2 < p < 2n^2$; such a prime exists for $n \geq 2$ by Bertrand's postulate. Encode a string as

$$f_x(z) = \sum_{i=0}^{n-1} x_{i+1} z^i \in \mathbb{F}_p[z].$$



Alice accepts iff $f_x(r) = f_y(r)$

Figure 2: A private-coin, one-way protocol for Equality. Bob's random point is sent along with the evaluation, so no shared randomness is assumed.

Bob chooses r uniformly in $\mathbb{F}_p$ and sends $(r, f_y(r))$. Alice evaluates $f_x(r)$ and compares. If $x = y$, the protocol always accepts. If $x \neq y$, then $h = f_x - f_y$ is a nonzero polynomial over $\mathbb{F}_p$: at least one coefficient is 1 or -1, hence remains nonzero modulo p. Since $\deg h \leq n - 1$,

$$\mathbb{P}[f_x(r) = f_y(r)] \leq \frac{n-1}{p} < \frac{1}{n}.$$

Both r and $f_y(r)$ use $\lceil \log p \rceil = O(\log n)$ bits. Randomization therefore reduces communication exponentially while preserving one-sided error.

Remark 4.1 (Public versus private coins)**.** If Alice and Bob share the random r, Bob sends only $f_y(r)$. The protocol above uses private randomness and explicitly communicates r. These are different models, although both have $O(\log n)$ cost here.

4.2 A random-prime checksum

There is a second construction that exposes a number-theoretic collision argument. Interpret x and y as integers in $[0, 2^n)$. Bob chooses a prime p uniformly from the primes at most $K = n^3$ and sends $(p, y \bmod p)$. Alice compares with $x \bmod p$.

For $x \neq y$, put $z = |x - y|$. A collision occurs exactly when $p \mid z$. Because $0 < z < 2^n$, it has fewer than n distinct prime divisors. Writing $\pi(K)$ for the number of primes at most K and using $\pi(K) = \Theta(K/\log K)$,

$$\mathbb{P}[x \equiv y \pmod p] \leq \frac{n}{\pi(n^3)} = O\left(\frac{\log n}{n^2}\right) = O(1/n).$$

The message again has $O(\log n)$ bits. This variant trades the polynomial root bound for a prime-divisor bound; generating a uniform random prime is an additional algorithmic task and should not be silently counted as one RAM operation.

5 Perfect matching through a determinant

Let $G = (L, R, E)$ be bipartite with $L = \{\ell_1, \dots, \ell_n\}$ and $R = \{r_1, \dots, r_n\}$. Introduce one formal variable x_{ij} per edge and form the Edmonds matrix [2]

$$M_G(i,j) = \begin{cases} x_{ij}, & (\ell_i, r_j) \in E, \\ 0, & (\ell_i, r_j) \notin E. \end{cases}$$

Theorem 5.1 (Edmonds' determinant criterion)**.** *G has a perfect matching if and only if the formal polynomial* $\det M_G$ *is not identically zero.*

Proof. Expanding the determinant gives

$$\det M_G = \sum_{\sigma \in S_n} \operatorname{sgn}(\sigma) \prod_{i=1}^{n} M_G(i, \sigma(i)).$$

Here S_n is the set of all permutations of $[n]$. The term indexed by σ is nonzero exactly when $\{(\ell_i, r_{\sigma(i)}) : i \in [n]\}$ is a perfect matching. Different permutations use different sets of edge variables, so their monomials are distinct and cannot cancel. Thus the expansion has a monomial precisely when a perfect matching exists. □

Choose a finite field $\mathbb{F}_q$ and independently assign every edge variable a uniform value from $\mathbb{F}_q$. Evaluate the resulting numeric determinant by Gaussian elimination.

Theorem 5.2 (Randomized matching detector)**.** *The detector uses $O(n^3)$ field operations. If G has no perfect matching, it always returns "no." If G has a perfect matching, it returns "no" with probability at most n/q.*

Proof. In a no-instance, theorem 5.1 says the determinant polynomial is zero, so every evaluation is zero. In a yes-instance it is a nonzero polynomial of total degree n. Schwartz–Zippel bounds the probability of a zero evaluation by n/q. Gaussian elimination computes one determinant in $O(n^3)$ field operations. □

Taking a prime $q \in (n^2, 2n^2)$ gives error below $1/n$ with $O(\log n)$-bit field elements. Deterministic polynomial-time bipartite matching algorithms are known; this application is valuable because it turns a combinatorial existence question into black-box PIT and exposes algebraic parallelism. It is not being presented as the fastest sequential matching algorithm.

6 Karp–Rabin pattern matching

Let a text $x = x_1 \cdots x_n \in \{0,1\}^n$ and a pattern $y = y_1 \cdots y_m \in \{0,1\}^m$ be given, with $m \le n$. A direct comparison at every alignment costs $O(nm)$ time. Karp and Rabin [4] compare short residues instead.

For a binary word $w = w_1 \cdots w_m$, write

$$[w]_2 = \sum_{j=1}^{m} w_j 2^{m-j}.$$

Let $X_i = x_i \cdots x_{i+m-1}$ be the window beginning at i. Choose a prime p uniformly from the primes at most $K = mn^3$, and use $H_p(w) = [w]_2 \bmod p$.

Karp–Rabin, Monte Carlo version

1. Compute $H_p(y)$, $H_p(X_1)$, and $a = 2^{m-1} \bmod p$.
2. For $i = 1, \ldots, n - m + 1$:
   (a) if $H_p(X_i) = H_p(y)$, report i as a match;
   (b) if another window remains, update its hash by

$$H_p(X_{i+1}) \equiv 2\big(H_p(X_i) - x_i a\big) + x_{i+m} \pmod{p}.$$

The update follows by removing the leading bit, shifting left, and appending the new bit. It uses only the previous residue; the exponentially large integer $[X_i]_2$ is never stored.

If $X_i \neq y$, then $0 < |[X_i]_2 - [y]_2| < 2^m$ and this difference has fewer than m distinct prime divisors. Hence

$$\mathbb{P}[H_p(X_i) = H_p(y)] \leq \frac{m}{\pi(mn^3)} = O\left(\frac{\log n}{n^3}\right),$$

where $m \leq n$ is used in the last estimate. A union bound over at most n windows gives

$$\mathbb{P}[\text{any false candidate}] = O\left(\frac{\log n}{n^2}\right) = o(1/n).$$

True matches always have equal hashes. Thus the decision version has no false negative; a reported location is correct except with the displayed probability. One may verify each reported candidate character by character to obtain certainty, but then the worst-case time depends on how many candidates are verified.

Since $p \leq mn^3 \leq n^4$, a field element fits in $O(\log n)$ bits. After p is chosen, preprocessing takes $O(m)$ and the scan takes $O(n)$ word-RAM operations, assuming $\Theta(\log n)$-bit modular arithmetic is constant time. In a bit-complexity model, the cost of modular arithmetic and random prime generation must be included. If the text itself arrives as a stream, the outgoing bit x_i must remain available, so the rolling recurrence also needs an m-bit circular buffer.

7 Multiset equality in a stream

Let $A = \{\{a_1, \ldots, a_n\}\}$ and $B = \{\{b_1, \ldots, b_n\}\}$ be multisets over $[n]$. Sorting compares them but stores the input; a frequency table takes $\Theta(n)$ counters, or $\Theta(n \log n)$ bits in the worst case. Even the special distinctness problem uses an n-bit seen vector. A permutation-invariant fingerprint due to Lipton [5] uses only logarithmic working space.

Associate with a multiset its root polynomial

$$f_A(z) = \prod_{i=1}^{n} (z - a_i), \qquad f_B(z) = \prod_{i=1}^{n} (z - b_i).$$

Over any field in which $1, \ldots, n$ are distinct, unique factorization shows that $f_A \equiv f_B$ exactly when $A = B$ as multisets: multiplicity in the multiset becomes multiplicity as a root.

7.1 The clean fixed-field algorithm

Fix a prime p with $n^2 < p < 2n^2$ and choose r uniformly in $\mathbb{F}_p$. Compute

$$\mathrm{FING}_r(A) = \prod_{i=1}^{n}(r - a_i) \pmod{p}, \qquad \mathrm{FING}_r(B) = \prod_{i=1}^{n}(r - b_i) \pmod{p}.$$

Theorem 7.1 (Streaming multiset equality)**.** *If $A = B$, the fingerprints are equal for every r. If $A \neq B$, then*

$$\mathbb{P}_r[\mathrm{FING}_r(A) = \mathrm{FING}_r(B)] \leq \frac{n-1}{p} < \frac{1}{n}.$$

The fingerprints can be computed in $O(n)$ modular operations and $O(\log n)$ working bits, in one pass over each stream.

Proof. Because $p > n$, the elements of $[n]$ remain distinct in $\mathbb{F}_p$. If $A \neq B$, the two monic root polynomials are distinct in $\mathbb{F}_p[z]$. Their leading terms cancel, so $h = f_A - f_B$ is nonzero of degree at most $n - 1$. The root bound gives the collision probability. During a stream, retain only r, p, a counter, and the running product; each occupies $O(\log n)$ bits. □

To check whether a stream $a_1, \ldots, a_n \in [n]$ contains every value exactly once, compare its fingerprint with $\prod_{i=1}^{n}(r - i) \bmod p$. This gives $O(n)$ time, $O(\log n)$ working space, and one-sided error below $1/n$.

7.2 Why one might also randomize the modulus

For the stated universe $[n]$, a fixed prime $p > n$ avoids every reduction ambiguity; randomizing p is unnecessary. The raw construction is useful, however, when elements come from a much larger integer universe and we want a small modulus. It introduces two distinct bad events that must not be conflated.

Regard $f_A - f_B$ first as a nonzero polynomial in $\mathbb{Z}[z]$. Choose a prime p uniformly from the primes in $[L, 2L]$ and reduce all coefficients modulo p. Then choose uniform $r \in \mathbb{F}_p$.

(1) **Bad modulus:** every coefficient difference becomes zero modulo p. It suffices that p divide one fixed nonzero integer coefficient c.

(2) **Bad evaluation:** the reduced difference remains nonzero, but the chosen r is one of its at most $n - 1$ roots.

For the original setting $a_i, b_i \in [n]$, every coefficient of either root polynomial has magnitude at most $(n+1)^n$, so we may take $0 < |c| \leq 2(n+1)^n$. Thus c has $O(n \log n)$ distinct prime divisors. The interval $[L, 2L]$ contains $\Theta(L/\log L)$ primes, while the second event has conditional probability at most n/L. With $L = \Theta((n \log n)^2)$,

$$\mathbb{P}[\text{collision}] \leq \mathbb{P}[\text{bad modulus}] + \mathbb{P}[\text{bad evaluation} \mid \text{good modulus}] = O(1/n).$$

This two-stage conditioning is essential: Schwartz–Zippel applies only after we know that reduction modulo p did not turn the formal difference into the zero polynomial.

8 How to design and audit a fingerprint

The examples differ in surface form, but a reliable analysis can be audited with the same questions.

Fingerprint design checklist

1. **What is the witness to inequality?** A nonzero row, a nonzero polynomial, a nonzero integer difference, or a changed root multiplicity.
2. **What exactly is random?** Coordinates, an evaluation point, a prime modulus, or both. State independence explicitly.
3. **Which bad choices hide the witness?** Count roots, divisors, or solutions of a linear equation.
4. **Which side is error-free?** Name false positives and false negatives.
5. **How large is the range?** Field size controls error but also controls message length, storage, and bit cost.
6. **What computation model supports the runtime?** Field operations, word-RAM operations, bit operations, and stream passes are not interchangeable.

Application	One trial	One-sided error	Typical parameter choice
Freivalds	$O(n^2)$ field ops	false accept $\leq 1/2$	repeat t times
PIT, degree d	one evaluation	false zero $\leq d/\lvert S\rvert$	$\lvert S\rvert \gg d$
Equality	$O(\log n)$ bits	false equal $< 1/n$	$p \asymp n^2$
Perfect matching	$O(n^3)$ field ops	false "no" $\leq n/q$	$q \asymp n^2$
Karp–Rabin	$O(n+m)$ word ops	false location $o(1/n)$	random $p \leq mn^3$
Multiset stream	$O(n)$ word ops, $O(\log n)$ bits	false equal $< 1/n$	fixed $p \asymp n^2$

Table 2: Guarantees established in this lecture. Asymptotic parameter choices suppress harmless constants and small-input handling.

9 Exercises

1. **Freivalds over a larger sample set.** Prove that choosing r uniformly from S^n gives false-accept probability at most $1/|S|$. What changes if the coordinates are not independent?
2. **Rectangular products.** Adapt Freivalds' verifier to $A \in \mathbb{F}^{a\times b}$, $B \in \mathbb{F}^{b\times c}$, and $C \in \mathbb{F}^{a\times c}$. Give the arithmetic cost and random-bit count.
3. **Tightness.** Construct A, B, C for which a false product is accepted by Freivalds with probability exactly $1/2$.
4. **Individual degree.** Suppose f has degree at most d_i in x_i. Prove $\mathbb{P}[f(r) = 0] \leq (\sum_i d_i)/|S|$ and compare it with the total-degree statement.
5. **Small fields.** Show that $x^p - x$ is nonzero in $\mathbb{F}_p[x]$ but induces the zero function on $\mathbb{F}_p$. Explain why evaluating it over $\mathbb{F}_{p^k}$ with $p^k > p$ can reveal nonzeroness.

6. **Equality parameters.** Choose p so that the polynomial protocol has error at most 2^{-40}. Express communication in terms of n and the target error δ.
7. **Shared randomness.** Reanalyse the Equality protocol when Alice and Bob share r. Which bits disappear from the message, and which randomness still exists in the experiment?
8. **Characteristic two.** Verify that the determinant criterion for bipartite perfect matching remains valid over a field of characteristic two, even though determinant signs disappear.
9. **A Las Vegas wrapper.** Modify Karp–Rabin to verify every reported candidate. Bound its expected time under the prime distribution used here, and identify an input with many genuine matches.
10. **Insertion–deletion streams.** A turnstile stream contains updates (a, Δ) to integer multiplicities. Propose a fingerprint using $\prod_a (r-a)^{\Delta}$ when negative exponents are well defined. State the event that must be excluded and how you would handle it.
11. **Randomness reuse.** Two different multiset comparisons use the same random r. Is the union bound still valid? Are the two collision events independent? Justify both answers.
12. **Audit a claim.** A note says: "A nonzero degree-d polynomial over $\mathbb{F}_p$ is nonzero at a random field point with probability at least $1 - d/p$." List the qualifications needed for this sentence to be informative and correct.

A Probability and prime-counting toolkit

A.1 Conditioning and independence

For $\mathbb{P}(B) > 0$, conditional probability is $\mathbb{P}(A \mid B) = \mathbb{P}(A \cap B)/\mathbb{P}(B)$. Splitting on B gives the law of total probability,

$$\mathbb{P}(A) = \mathbb{P}(A \mid B)\mathbb{P}(B) + \mathbb{P}(A \mid B^c)\mathbb{P}(B^c).$$

Random variables are independent when conditioning on one does not change the distribution of the others. This is the step used in the Schwartz–Zippel proof: after fixing $r_1, \ldots, r_{m-1}$, the final coordinate r_m remains uniform in S.

A.2 Union bound and amplification

For arbitrary events $E_1, \ldots, E_k$,

$$\mathbb{P}\left[\bigcup_{i=1}^{k} E_i\right] \le \sum_{i=1}^{k} \mathbb{P}(E_i).$$

No independence is required. This is why Karp–Rabin may reuse one prime across all windows: the collision events can be dependent and the union bound still applies.

If independent trials each fail with probability at most ε, all t fail with probability at most ε^t. More generally, if a trial succeeds with probability at least q, then

$$\mathbb{P}[\text{no success in } t \text{ independent trials}] \le (1-q)^t \le e^{-qt}.$$

A.3 Counting prime divisors

If a nonzero integer z has k distinct prime divisors, then $|z| \geq 2^k$. Hence

$$\omega(z) \leq \log_2 |z|,$$

where $\omega(z)$ counts distinct prime divisors. This deliberately coarse bound is enough for the checksum and rolling-hash analyses.

Let $\pi(N)$ be the number of primes at most N. The Prime Number Theorem states

$$\pi(N) \sim \frac{N}{\ln N}.$$

We use only the coarser consequences $\pi(N) = \Theta(N/\log N)$ and $\pi(2N) - \pi(N) = \Theta(N/\log N)$ for sufficiently large N. Asymptotic algorithm statements can absorb the finitely many smaller inputs into their constants. Bertrand's postulate supplies the simpler fact that for every integer $N > 1$ there is a prime strictly between N and $2N$.

B Finite fields and polynomial algebra

B.1 Fields and prime fields

A *field* $\mathbb{F}$ has addition and multiplication, additive and multiplicative identities $0, 1$, additive inverses, and a multiplicative inverse for every nonzero element; the usual associative, commutative, and distributive laws hold. The integers modulo a prime p form the field

$$\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}.$$

Modulo a composite number, nonzero zero divisors may exist, so division by a nonzero residue is not always valid. This is why the lecture chooses prime moduli when it invokes field algebra.

Every finite field has order p^k for a prime p and integer $k \geq 1$, and for each prime power there is, up to isomorphism, a unique finite field $\mathbb{F}_{p^k}$. One construction is

$$\mathbb{F}_{p^k} \cong \mathbb{F}_p[z]/(g(z)),$$

where g is irreducible of degree k. Extension fields let us create an evaluation domain larger than a small base field without changing its characteristic.

B.2 Polynomial rings and division

The polynomial ring $\mathbb{F}[x]$ consists of formal finite sums $a_0 + a_1 x + \cdots + a_d x^d$ with coefficients in $\mathbb{F}$. If $g \neq 0$, polynomial division gives unique $q, r \in \mathbb{F}[x]$ such that

$$f = qg + r, \qquad \deg r < \deg g.$$

Taking $g = x - a$ yields the factor theorem: $f(a) = 0$ if and only if $(x - a)$ divides f. Repeating this observation proves the univariate root bound used throughout the lecture.

A nonconstant polynomial is *irreducible* if it cannot be written as a product of two lower-degree nonconstant polynomials. Quotienting by an irreducible polynomial is the polynomial analogue of reducing integers modulo a prime.

B.3 Formal polynomials versus polynomial functions

An element of $\mathbb{F}[x_1, \ldots, x_m]$ is a coefficient vector with algebraic structure; evaluation turns it into a function $\mathbb{F}^m \to \mathbb{F}$. Over an infinite field, two formal polynomials induce the same function only if they are equal. Over a finite field this need not hold. In particular,

$$x^p - x = 0 \qquad \text{for every } x \in \mathbb{F}_p$$

by Fermat's little theorem, although $x^p - x$ is not the zero polynomial. Schwartz–Zippel avoids confusion by beginning with a nonzero *formal* polynomial and sampling from a set whose size is compared with its degree.

C Bibliographic notes

Freivalds' verifier appeared in his 1977 IFIP paper, although it is often dated 1979 in lecture notes because of related later publications. The multivariate root bound is conventionally called Schwartz–Zippel, while DeMillo and Lipton obtained a closely related result earlier; the combined name DeMillo–Lipton–Schwartz–Zippel is therefore also used. Edmonds' matrix in this lecture is the bipartite determinant construction, not the skew-symmetric Tutte matrix used for general-graph matching. Lipton's permutation-invariant product fingerprint was circulated as the Princeton technical report *Fingerprinting Sets*.

References

[1] R. A. DeMillo and R. J. Lipton. A probabilistic remark on algebraic program testing. *Information Processing Letters*, 7(4):193–195, 1978. doi:10.1016/0020-0190(78)90067-4.

[2] J. Edmonds. Systems of distinct representatives and linear algebra. *Journal of Research of the National Bureau of Standards B*, 71B(4):241–245, 1967.

[3] R. Freivalds. Probabilistic machines can use less running time. In *Information Processing 77*, pages 839–842. North-Holland, 1977.

[4] R. M. Karp and M. O. Rabin. Efficient randomized pattern-matching algorithms. *IBM Journal of Research and Development*, 31(2):249–260, 1987. doi:10.1147/rd.312.0249.

[5] R. J. Lipton. Fingerprinting sets. Technical Report CS-TR-212-89, Princeton University, 1989.

[6] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM*, 27(4):701–717, 1980. doi:10.1145/322217.322225.

[7] A. C.-C. Yao. Some complexity questions related to distributive computing. In *Proceedings of the 11th ACM Symposium on Theory of Computing*, pages 209–213, 1979. doi:10.1145/800135.804414.

[8] R. Zippel. Probabilistic algorithms for sparse polynomials. In *EUROSAM 1979*, Lecture Notes in Computer Science 72, pages 216–226. Springer, 1979.