# Mathematical Background

# for Information Theory and Applications in Computer Science

Review with some guided introductions

Fall 2026 • Marshall Ball

**Purpose**

This packet refreshes the mathematics used throughout the course, and introduces a few things you may not have seen before.
I expect you to be comfortable with finite probability, logarithms and elementary calculus, counting, basic linear algebra, and proof reading/writing.
Some exercises are included for practice.

# Contents

# 1 Notation and conventions

## 1.1 Sets, strings, and coordinates

| Notation | Meaning |
|---|---|
| $[n]$ | The set $\{1, 2, \ldots, n\}$. |
| $\lvert S\rvert$ | Cardinality of a finite set $S$; length of a string when the argument is a string. |
| $x^n = (x_1, \ldots, x_n)$ | A length-$n$ sequence. This is notation, not exponentiation. |
| $x_{<i}$ | The prefix $(x_1, \ldots, x_{i-1})$. More generally, $x_A = (x_i)_{i \in A}$. |
| $\{0,1\}^n$ | The set of binary strings of length $n$. |
| $\mathrm{wt}(x)$ | Hamming weight: the number of nonzero coordinates of $x$. |
| $\Delta_H(x, y)$ | Hamming distance: the number of coordinates where $x$ and $y$ differ. For vectors over a field, $\Delta_H(x, y) = \mathrm{wt}(x - y)$. |
| $\mathbf{1}_E$ | Indicator of event $E$: it equals 1 on $E$ and 0 otherwise. |
| $\oplus$ | Addition modulo two, or bitwise XOR when applied to binary strings. |

Uppercase letters such as $X, Y, Z$ denote random variables; lowercase letters $x, y, z$ denote values. Script letters such as $\mathcal{X}$ denote alphabets or sets. We write $X \sim P$ when $X$ has distribution $P$, and $P_X$ for the distribution induced by $X$.

## 1.2 Probability notation

| Expression | Reading |
|---|---|
| $\Pr[X = x]$ | Probability that $X$ equals $x$. |
| $\Pr_{x \leftarrow P}[E(x)]$ | Sample $x$ from $P$ and take the probability that $E(x)$ occurs. Also written $\Pr_{X \sim P}[E(X)]$; a subscript always names the source of randomness. |
| $U_S$ (and $U_n$) | The uniform distribution on a finite set $S$ (and on $\{0,1\}^n$, respectively). |
| $\mathrm{Bern}(p)$ | The Bernoulli distribution: it equals 1 with probability $p$ and 0 otherwise. |
| $\mathbb{E}[f(X)]$ | Expected value of $f(X)$. |
| $X \perp Y$ | $X$ and $Y$ are independent. |
| $X \perp Z \mid Y$ | $X$ and $Z$ are conditionally independent given $Y$. |
| $X \to Y \to Z$ | A Markov chain: conditioned on $Y$, the variable $Z$ carries no further dependence on $X$. |
| $X_1, \ldots, X_n \overset{\text{i.i.d.}}{\sim} P$ | Independent random variables with common distribution $P$. |

## 1.3 Logarithms and zero-probability conventions

Throughout,

$$\log x = \log_2 x, \qquad \log(ab) = \log a + \log b, \qquad \log(a/b) = \log a - \log b.$$

When a zero-probability term appears inside a weighted sum, use its limiting value. In particular,

$$0 \log 0 = 0, \qquad 0 \log \frac{0}{q} = 0 \quad (q > 0),$$

and an entropy summand $p \log(1/p)$ contributes 0 when $p = 0$. By contrast, $p \log(p/0) = +\infty$ when $p > 0$.

### 1.4 Asymptotic notation

For nonnegative functions $f, g$ (with $g(n) > 0$ eventually when a ratio is used):

- $f(n) = O(g(n))$ means that there are constants $C > 0$ and $n_0$ such that $f(n) \le C g(n)$ for every $n \ge n_0$.
- $f(n) = \Omega(g(n))$ means that there are constants $c > 0$ and $n_0$ such that $f(n) \ge c g(n)$ for every $n \ge n_0$.
- $f(n) = \Theta(g(n))$ means both $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.
- $f(n) = o(g(n))$ means $f(n)/g(n) \to 0$.
- $\widetilde{O}(g(n))$ usually suppresses polylogarithmic factors: $f(n) = \widetilde{O}(g(n))$ if $f(n) = O\big(g(n) \log^C n\big)$ for some constant $C \ge 0$.[1]

Thus $2^{-\Omega(n)}$ means at most $2^{-cn}$ for some constant $c > 0$ and all sufficiently large $n$.

In this packet, *with high probability* means probability $1 - o(1)$ as $n \to \infty$. Some sources require a specified inverse-polynomial failure bound, so check the convention. In cryptographic contexts, a function $\mu(n)$ is *negligible* if, for every constant $c > 0$, one has $\mu(n) < n^{-c}$ for all sufficiently large $n$.

## 2 Discrete probability

### 2.1 Distributions, joint distributions, and marginals

A probability distribution $p$ on a finite alphabet $\mathcal{X}$ assigns numbers $p(x) \ge 0$ satisfying $\sum_x p(x) = 1$. Its support is

$$\operatorname{supp}(p) = \{x \in \mathcal{X} : p(x) > 0\}.$$

For jointly distributed $(X, Y)$, the joint probability mass function is $p_{X,Y}(x, y) = \Pr[X = x, Y = y]$. The marginals are obtained by summing out the other variable:

$$p_X(x) = \sum_y p_{X,Y}(x, y), \qquad p_Y(y) = \sum_x p_{X,Y}(x, y).$$

[1] Some authors use "soft-$O$" more loosely to suppress specified subpolynomial factors, such as $2^{\sqrt{\log n}} = n^{o(1)}$. Check the convention in the source you are reading.

**Example: a binary symmetric channel**

Let $X \sim \mathrm{Bern}(1/2)$ and let $N \sim \mathrm{Bern}(p)$ be independent. Define $Y = X \oplus N$. Then $Y$ is uniform, but $X$ and $Y$ are generally not independent: they agree with probability $1 - p$.

$$p_X(0) = p_X(1) = \frac{1}{2}, \qquad p_Y(0) = \frac{1}{2}(1-p) + \frac{1}{2}p = \frac{1}{2} = p_Y(1),$$

$$p_{X,Y}(0,0) = p_{X,Y}(1,1) = \frac{1-p}{2}, \qquad p_{X,Y}(0,1) = p_{X,Y}(1,0) = \frac{p}{2}.$$

(We will revisit this example when discussing noisy channels.)

## 2.2 Conditioning, total probability, and Bayes

For an event $B$ of positive probability,

$$\Pr[A \mid B] = \frac{\Pr[A \cap B]}{\Pr[B]}.$$

For $p_Y(y) > 0$,

$$p_{X|Y}(x \mid y) = \frac{p_{X,Y}(x,y)}{p_Y(y)}.$$

The law of total probability and Bayes' rule are

$$\Pr[A] = \sum_{y : p_Y(y) > 0} \Pr[A \mid Y = y] p_Y(y),$$

$$\Pr[X = x \mid Y = y] = \frac{\Pr[Y = y \mid X = x] \Pr[X = x]}{\Pr[Y = y]} = \frac{\Pr[Y = y \wedge X = x]}{\Pr[Y = y]}.$$

The tower property:

$$\mathbb{E}[Z] = \mathbb{E}\big[\mathbb{E}[Z \mid Y]\big].$$

Here $\mathbb{E}[Z \mid Y]$ is itself a random variable: it is the function of $Y$ whose value at $y$ is $\sum_z z\, p_{Z|Y}(z \mid y)$. All sums over conditional distributions omit zero-probability conditioning events. If $W$ is a function of $Y$, the more general tower identity is

$$\mathbb{E}\big[\mathbb{E}[Z \mid Y] \mid W\big] = \mathbb{E}[Z \mid W].$$

## 2.3 Functions, channels, and product distributions

If $Y = f(X)$, its induced distribution is obtained by summing over a *fiber* (or, preimages):

$$p_Y(y) = \sum_{x : f(x) = y} p_X(x).$$

Equality of distributions is not equality of random variables: two independent unbiased bits $X, X'$ have the same distribution but differ with probability $1/2$. Likewise, marginals do not determine a joint distribution: $(X, X)$ and $(X, X')$ have the same one-coordinate marginals but different joint distributions.

A *channel* $K(y \mid x)$ is simply a probability distribution on outputs for each fixed input $x$. Thus $K(y \mid x) \geq 0$ and $\sum_y K(y \mid x) = 1$. Sampling $X \sim P$ and then $Y \sim K(\cdot \mid X)$ gives

$$p_{X,Y}(x,y) = P(x)K(y \mid x), \qquad (PK)(y) = \sum_x P(x)K(y \mid x).$$

A deterministic function is the special case $K(y \mid x) = \mathbf{1}_{\{y=f(x)\}}$.

Repeated conditioning gives the probability chain rule, on every positive-probability sequence:

$$p_{X_1,\dots,X_n}(x^n) = \prod_{i=1}^{n} p_{X_i|X_{<i}}(x_i \mid x_{<i}).$$

There is no independence assumption here. Under independence this becomes $\prod_i p_{X_i}(x_i)$; with a common marginal $P$ we call it the *product distribution* $P^{\otimes n}$. Taking a logarithm turns either factorization into a sum. This elementary identity will underlie the course's information chain rules.

## 2.4 Independence and Markov structure

Random variables $X$ and $Y$ are *independent*, $X \perp Y$, when

$$p_{X,Y}(x,y) = p_X(x)p_Y(y) \quad \text{for every } x, y.$$

More generally, $X_1, \dots, X_k$ are *jointly independent* if

$$p_{X_1,\dots,X_k}(x_1,\dots,x_k) = p_{X_1}(x_1)\cdots p_{X_k}(x_k) \quad \text{for every } x_1,\dots,x_k.$$

$X_1, \dots, X_k$ are *pairwise independent* if $X_i \perp X_j$ for all $i \neq j$.

**Pairwise independence is not mutual independence**

Three variables can be independent in every pair while satisfying a deterministic relation jointly. A standard example is $Z = X \oplus Y$ for independent unbiased bits $X, Y$. Every pair among $X, Y, Z$ is independent, but the three variables are not jointly independent.

Conditional independence $X \perp Z \mid Y$ means

$$p_{X,Z|Y}(x,z \mid y) = p_{X|Y}(x \mid y)p_{Z|Y}(z \mid y)$$

for every $y$ with positive probability. The notation $X \to Y \to Z$ denotes this Markov condition. One common generative form is

$$p(x,y,z) = p_X(x)K(y \mid x)L(z \mid y).$$

Here $Y$ is generated from $X$ through a channel $K$, and $Z$ is generated from $Y$ through a channel $L$; once $Y$ is known, $X$ provides no additional information about $Z$.

## 2.5 Expectation, indicators, and variance

For a finite random variable $X$ and a real-valued function $f$,

$$\mathbb{E}[f(X)] = \sum_x p_X(x)f(x).$$

Linearity of expectation requires no independence:

$$\mathbb{E}\left[\sum_{i=1}^{n} a_i X_i\right] = \sum_{i=1}^{n} a_i \mathbb{E}[X_i].$$

Indicator random variables can be very helpful, a frequent template: If $N = \sum_i \mathbf{1}_{E_i}$ counts how many events occur, then

$$\mathbb{E}[N] = \sum_i \mathbb{E}[\mathbf{1}_{E_i}] = \sum_i \Pr[E_i].$$

Variance is

$$\mathrm{Var}(X) = \mathbb{E}[(X - \mathbb{E}X)^2] = \mathbb{E}[X^2] - (\mathbb{E}X)^2.$$

For real-valued variables, define

$$\mathrm{Cov}(X, Y) = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y].$$

If $X \perp Y$, then $\mathbb{E}[XY] = \sum_{x,y} xy\, p_X(x) p_Y(y) = \mathbb{E}[X]\mathbb{E}[Y]$, so $\mathrm{Cov}(X, Y) = 0$. The converse fails: if $X$ is uniform on $\{-1, 0, 1\}$ and $Y = X^2$, then $\mathbb{E}[XY] = \mathbb{E}[X^3] = 0 = \mathbb{E}[X]\mathbb{E}[Y]$, yet $Y$ is a function of $X$. Expanding a square gives

$$\mathrm{Var}\left(\sum_i X_i\right) = \sum_i \mathrm{Var}(X_i) + 2\sum_{i<j} \mathrm{Cov}(X_i, X_j).$$

Independence of each pair makes its covariance zero. Thus *pairwise independence is enough* for variance additivity:

$$\mathrm{Var}\left(\sum_i X_i\right) = \sum_i \mathrm{Var}(X_i).$$

## 2.6 The basic probability inequalities

**Union bound**

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

$$\Pr\left[\bigcup_{i=1}^{m} E_i\right] \le \sum_{i=1}^{m} \Pr[E_i].$$

No independence is required.

**Markov and Chebyshev**

If $Z \ge 0$ and $a > 0$, then

$$\Pr[Z \ge a] \le \frac{\mathbb{E}Z}{a}.$$

Applying this to $Z = (X - \mathbb{E}X)^2$ with $a = t^2$ gives Chebyshev's inequality: for every $t > 0$,

$$\Pr[|X - \mathbb{E}X| \ge t] \le \frac{\mathrm{Var}(X)}{t^2}.$$

If $X_1, \ldots, X_n$ are i.i.d. with mean $\mu$ and variance $\sigma^2$, Chebyshev gives (weak law of large numbers):

$$\Pr\left[\left|\frac{1}{n}\sum_{i=1}^{n} X_i - \mu\right| \ge \varepsilon\right] \le \frac{\sigma^2}{n\varepsilon^2}.$$

### 2.7 Exponential tail bounds

Some later arguments need a stronger bound than Chebyshev. For mutually independent $X_i \in [0,1]$ and $\varepsilon > 0$, with $\overline{X} = n^{-1}\sum_i X_i$, Hoeffding's inequality states

$$\Pr\bigl[|\overline{X} - \mathbb{E}\overline{X}| \geq \varepsilon\bigr] \leq 2e^{-2n\varepsilon^2}.$$

## 3 Logarithms, convexity, and analytic inequalities

### 3.1 Base-two exponential scale

Information-theoretic arguments often compare a number of candidates, such as $2^{nR}$, against an exponentially small probability, such as $2^{-n\alpha}$. Multiplication becomes addition in the exponent:

$$2^{nR}2^{-n\alpha} = 2^{-n(\alpha - R)}.$$

(The sign of $\alpha - R$ determines whether the expression vanishes or grows.)

### 3.2 Convexity and Jensen's inequality

A function $\varphi$ on a real interval is convex if

$$\varphi(\lambda x + (1-\lambda)y) \leq \lambda\varphi(x) + (1-\lambda)\varphi(y)$$

for $0 \leq \lambda \leq 1$. It is concave when the inequality is reversed. Important examples are:

- $x \mapsto -\log x$ is strictly convex on $(0,\infty)$;
- $x \mapsto \log x$ is strictly concave on $(0,\infty)$;
- $x \mapsto x^2$ and $x \mapsto x\log x$ are convex on $[0,\infty)$, with $0\log 0 = 0$;
- $x \mapsto \sqrt{x}$ is concave on $[0,\infty)$.

**Jensen's inequality**

For a convex $\varphi$ and a finite-valued real random variable $X$ taking values in its domain,

$$\varphi(\mathbb{E}X) \leq \mathbb{E}[\varphi(X)].$$

If $\varphi$ is strictly convex, equality holds exactly when $X$ is constant with probability one. Values assigned probability zero are irrelevant. For a concave $\varphi$ the inequality reverses: $\varphi(\mathbb{E}X) \geq \mathbb{E}[\varphi(X)]$.

For example:

$$\mathbb{E}[-\log Z] \geq -\log \mathbb{E}[Z] \qquad (Z > 0 \text{ almost surely}).$$

Equivalently, for weights $\lambda_i \geq 0$ summing to one,

$$\varphi\left(\sum_i \lambda_i x_i\right) \leq \sum_i \lambda_i \varphi(x_i).$$

### 3.3 A little bit of calculus

The derivatives

$$\frac{d}{dx}\log x = \frac{1}{x\ln 2}, \qquad \frac{d^2}{dx^2}(x\log x) = \frac{1}{x\ln 2} \quad (x>0)$$

explain several of the preceding curvature claims. For a twice-differentiable function on an interval, $\varphi'' \geq 0$ implies convexity. A differentiable convex function lies above each tangent line:

$$\varphi(y) \geq \varphi(x) + \varphi'(x)(y-x).$$

For example, the tangent bound for the concave natural logarithm at 1 is $\ln u \leq u-1$ for $u>0$. Consequently $1-v \leq e^{-v}$ for every real $v$: substitute $u = 1-v$ when $v<1$, and note that the left side is nonpositive when $v \geq 1$. A typical use is $(1-p)^n \leq e^{-pn}$ for $0 \leq p \leq 1$, which turns a product of near-one factors into an exponential. (We will at most use derivatives and such one-variable inequalities, no serious analysis.)

### 3.4 Cauchy–Schwarz and norm comparisons

For real sequences $(a_i)$ and $(b_i)$,

$$\left(\sum_i a_i b_i\right)^2 \leq \left(\sum_i a_i^2\right)\left(\sum_i b_i^2\right).$$

Applying this with $|a_i|$ in place of $a_i$ and $b_i = 1$ gives

$$\left(\sum_{i=1}^m |a_i|\right)^2 \leq m\sum_{i=1}^m a_i^2.$$

Looking ahead, this $\ell_1$ vs $\ell_2$ comparison can convert a collision or second-moment estimate into a bound on statistical distance (total variation distance).

## 4 Counting and asymptotics

### 4.1 Binomial and multinomial counting

The number of binary strings of length $n$ with Hamming weight $k$ (exactly $k$ ones) is

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

For nonnegative counts $n_1 + \cdots + n_q = n$, the number of strings over a $q$-symbol alphabet with exactly $n_j$ occurrences of symbol $j$ is

$$\binom{n}{n_1,\ldots,n_q} = \frac{n!}{n_1!\cdots n_q!}.$$

A frequently useful elementary bound is

$$\binom{n}{k} \leq \frac{n^k}{k!} \leq \left(\frac{en}{k}\right)^k, \qquad 1 \leq k \leq n.$$

(The inequality follows from $k! \geq (k/e)^k$, which in turn follows from $e^k = \sum_j k^j/j! \geq k^k/k!$.)

## 4.2 Stirling's approximation

For this course,

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{O(1/n)},$$

so, with base-two logs,

$$\log(n!) = n\log n - (\log e)n + O(\log n).$$

## 4.3 The binary entropy estimate

Define the binary entropy function

$$h_2(p) = -p\log p - (1-p)\log(1-p), \qquad 0 \le p \le 1,$$

with $0\log 0 = 0$. Stirling implies that for fixed $p \in (0,1)$ and $k = \lfloor pn \rfloor$,

$$\binom{n}{k} = 2^{nh_2(p)+O(\log n)}.$$

Rounding $pn$ down changes the exponent by only $O(1)$ for fixed $p \in (0,1)$. To see the exponent before doing the full Stirling calculation, set $\theta = k/n$: the powers $n^n/(k^k(n-k)^{n-k})$ simplify to $2^{nh_2(\theta)}$.

Also, $h_2(p) = h_2(1-p)$, and $h_2'(p) = \log((1-p)/p)$ on $(0,1)$, so $h_2$ increases from $h_2(0) = 0$ to $h_2(1/2) = 1$ and then decreases symmetrically.

## 4.4 Hamming balls: an information motivated counting review

For $x, y \in \{0,1\}^n$, the Hamming distance $\Delta_H(x,y)$ counts differing coordinates. For an integer $0 \le t \le n$, define

$$B(y,t) = \{x \in \{0,1\}^n : \Delta_H(x,y) \le t\}.$$

A string at distance $j$ from $y$ is obtained by choosing exactly $j$ coordinates to flip. Hence the volume is independent of the center:

$$V_2(n,t) := |B(y,t)| = \sum_{j=0}^{t}\binom{n}{j}.$$

For example, $V_2(4,1) = 1 + 4 = 5$. Over an alphabet of size $q$, each changed coordinate has $q-1$ replacement choices, giving $V_q(n,t) = \sum_{j=0}^{t}\binom{n}{j}(q-1)^j$.

**An exponential upper bound**

For $0 \le \rho \le 1/2$,

$$V_2(n, \lfloor \rho n \rfloor) \le 2^{nh_2(\rho)}.$$

*Proof.* The case $\rho = 0$ is immediate. For $0 < \rho \le 1/2$, independently make each bit 1 with probability $\rho$. A particular string of weight $j \le \rho n$ has probability

$$\rho^j(1-\rho)^{n-j} = (1-\rho)^n\left(\frac{\rho}{1-\rho}\right)^j \ge \rho^{\rho n}(1-\rho)^{(1-\rho)n} = 2^{-nh_2(\rho)}.$$

The inequality uses $\rho/(1-\rho) \le 1$ together with $j \le \rho n$; the final equality is the definition of $h_2(\rho)$ in exponential form (take logarithms to check it). All strings of weight at most $\lfloor \rho n \rfloor$ together have probability at most one, so their number is at most $2^{nh_2(\rho)}$. □

For fixed $0 < \rho \le 1/2$, the outermost layer and Stirling also give

$$\log V_2(n, \lfloor \rho n \rfloor) = n h_2(\rho) + O(\log n).$$

Indeed, $V_2(n,t) \ge \binom{n}{t}$, while the preceding upper bound gives the other direction. The ball can contain exponentially many strings and still occupy an exponentially small fraction of the cube when $\rho < 1/2$:

$$\Pr_{X \sim U_n}[X \in B(y, \lfloor \rho n \rfloor)] \le 2^{-n(1-h_2(\rho))}.$$

The restriction $\rho \le 1/2$ matters. For fixed $\rho > 1/2$, the ball contains at least half the cube for all sufficiently large $n$; $h_2(\rho) < 1$ is then the wrong volume exponent.

**Coding preview.** Call a set $C \subseteq \{0,1\}^n$ a *code*, and its elements *codewords*. If different codewords are at distance at least $2t+1$, their radius-$t$ balls are disjoint: a point in two balls would give distance at most $2t$ between the centers by the triangle inequality. Counting disjoint balls therefore yields

$$|C|\, V_2(n,t) \le 2^n.$$

This is the sphere-packing bound. Its proof uses only counting and a metric; no coding-theory background is needed. Problem 10 asks you to reconstruct and apply these arguments. For more, see [GRS25, Section 3.3.1].

## 4.5 Empirical distributions (types)

For a sequence $x^n$ over an alphabet $\mathcal{X}$, its *empirical distribution*, or *type*, is

$$\widehat{P}_{x^n}(a) = \frac{1}{n} |\{i : x_i = a\}|.$$

If $|\mathcal{X}| = q$, there are at most $(n+1)^q$ possible empirical distributions, since each symbol count is an integer in $\{0, \dots, n\}$. For a type $Q$, its *type class* is

$$T_Q = \{x^n : \widehat{P}_{x^n} = Q\}.$$

If the corresponding symbol counts are $n_1, \dots, n_q$, then

$$|T_Q| = \binom{n}{n_1, \dots, n_q}.$$

## 4.6 The probabilistic method

Two elementary templates appear frequently:

**Positive probability implies existence**

If a random object avoids all bad events with positive probability, then at least one good deterministic object exists. In particular, if

$$\sum_i \Pr[E_i] < 1,$$

then the union bound guarantees an outcome outside every $E_i$.

**Expectation below one implies a zero**

If $N$ is a nonnegative integer-valued random variable and $\mathbb{E}N < 1$, then some outcome has $N = 0$.

# 5 Linear algebra, finite fields, and polynomials

We review familiar linear algebra over a different field, then use a subspace as a first example of a code. The coding vocabulary below is defined as it appears; the work is solving linear equations, finding bases, and counting solutions.

## 5.1 Finite fields and vector spaces

The field $\mathbb{F}_2 = \{0, 1\}$ uses addition and multiplication modulo two. More generally, $\mathbb{F}_q$ denotes a finite field of size $q$ (where $q$ is a prime power). Familiar linear-algebra notions carry over:

- linear combinations, span, basis, and dimension;
- linear maps and matrices;
- kernel and image;
- rank–nullity: $\dim \ker A + \operatorname{rank} A = \dim(\text{domain})$.

Over $\mathbb{F}_2$, subtraction and addition are the same operation. Gaussian elimination works because every nonzero field element is invertible.

For prime $p$, $\mathbb{F}_p$ is arithmetic modulo $p$. For $q = p^r$ with $r > 1$, $\mathbb{F}_q$ is *not* arithmetic modulo $q$: for instance, $2 \cdot 2 = 0$ modulo 4, so integers modulo 4 do not form a field. Extension fields can be represented using polynomials modulo an irreducible polynomial. All computations requested here use prime fields; the extension-field construction is only needed to interpret one later hashing example.

## 5.2 Counting solutions of a linear system

Let $A : \mathbb{F}_q^n \to \mathbb{F}_q^r$ be linear and have rank $s$. Rank–nullity and the fact that a $d$-dimensional space has $q^d$ vectors give

$$|\ker A| = q^{n-s}, \qquad |\operatorname{im} A| = q^s.$$

Every nonempty fiber $\{x : Ax = y\}$ is a translate of the kernel: if $Ax_0 = y$, then $Ax = y$ exactly when $x - x_0 \in \ker A$. Therefore every attainable right-hand side has exactly $q^{n-s}$ solutions. In particular, a uniform input has a uniform image *on the image of* $A$, not necessarily on the whole codomain.

This observation has two uses below: it counts messages or syndromes in a code, and it counts seeds that produce prescribed hash outputs.

## 5.3 The same subspace, as a (not necessarily good) linear code

A *linear code* is simply a subspace $C \leq \mathbb{F}_q^n$. Suppose $\dim C = k \geq 1$. Put a basis of $C$ into the rows of a $k \times n$ matrix $G$. Then

$$C = \{mG : m \in \mathbb{F}_q^k\}, \qquad |C| = q^k.$$

Here vectors $m$ and codewords are (linear combinations of) rows. The map $m \mapsto mG$ is injective because the rows are linearly independent. Coding terminology calls $m$ the *message*, $mG$ its *codeword*, and $G$ a *generator matrix*.

The same $C$ can be specified by $n - k$ independent homogeneous linear equations. If their coefficient matrix is $H \in \mathbb{F}_q^{(n-k)\times n}$, then

$$C = \{c : Hc^\top = 0\}.$$

This $H$ is called a *parity-check matrix*. To verify that proposed matrices $G, H$ describe the same code, check

$$\operatorname{rank} G = k, \qquad \operatorname{rank} H = n - k, \qquad HG^{\top} = 0.$$

The last identity puts the row space of $G$ inside the kernel of $H$; rank–nullity gives both spaces dimension $k$, so they are equal. Conversely, a basis for the solutions of $Gz = 0$ supplies the rows of a suitable $H$.

The *minimum distance* of a nontrivial linear code is

$$d(C) = \min_{\substack{c,c' \in C \\ c \neq c'}} \Delta_H(c, c') = \min_{0 \neq c \in C} \operatorname{wt}(c).$$

Indeed, $\Delta_H(c, c') = \operatorname{wt}(c - c')$, differences of codewords stay in $C$, and every nonzero codeword is its own difference from zero.

## 5.4 A syndrome is just the value of a linear map

For a received vector $y$, its *syndrome* is $Hy^{\top}$. If $y = c + e$ with $c \in C$, then

$$Hy^{\top} = Hc^{\top} + He^{\top} = He^{\top}.$$

The codeword disappears, leaving equations about the error $e$. A zero syndrome means only that $y \in C$; it does not rule out an error that changed one codeword into another.

**A three-bit example**

Over $\mathbb{F}_2$, take

$$H = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \qquad G = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}.$$

The equations say $x_1 = x_2 = x_3$, so $C = \{000, 111\}$ has dimension 1, size 2, and minimum distance 3. The three single-bit errors have syndromes equal to the columns of $H$: $(1,0)^{\top}$, $(1,1)^{\top}$, and $(0,1)^{\top}$. They are distinct and nonzero, so the syndrome identifies the error when at most one bit was changed. This is ordinary matrix multiplication, interpreted as error correction.

## 5.5 Polynomials over a field

Two useful algebraic facts:

**Root bound**

A nonzero polynomial of degree at most $d$ over a field has at most $d$ roots.

**Interpolation**

Given distinct field elements $\alpha_1, \dots, \alpha_k$ and arbitrary values $y_1, \dots, y_k$, there is a unique polynomial $p$ of degree less than $k$ satisfying $p(\alpha_i) = y_i$ for all $i$.

The uniqueness statement follows immediately from the root bound. Existence is given by the Lagrange formula

$$p(x) = \sum_{i=1}^{k} y_i \prod_{j \neq i} \frac{x - \alpha_j}{\alpha_i - \alpha_j}.$$

Evaluation is a linear map from coefficient vectors to value vectors. For $k$ distinct points, its matrix is the Vandermonde matrix $V_{ij} = \alpha_i^{j-1}$, which is invertible: a vector in its kernel would describe a polynomial of degree less than $k$ with $k$ roots. This recasts interpolation as linear algebra and will also explain limited-independence constructions.

# 6 Pairwise-independent hashing: a tutorial

## 6.1 A random function with a short description

Suppose we want to assign each of many possible inputs a label in $[M]$. A completely random function assigns independent uniform labels to every input. Its full table has one entry per input, which may be far too large to store. A *hash family* replaces that table by a short random description of a deterministic function.

Formally, let $\mathcal{S}$ be a finite seed set and let

$$h : \mathcal{S} \times \mathcal{D} \longrightarrow \mathcal{R}, \qquad h_s(x) := h(s, x).$$

Choose $S$ uniformly in $\mathcal{S}$, once. Thereafter $h_S$ is the chosen function. The same input always has the same output; *we do not resample a seed at each evaluation.* The inputs $x, x'$ in the definition below are fixed, while the seed is random.

**Definition: pairwise-independent hashing**

Assume $|\mathcal{D}| \geq 2$ and $|\mathcal{R}| = M$. The seeded family is *pairwise independent*, also called *strongly 2-universal*, if for every distinct $x, x' \in \mathcal{D}$ and every $u, v \in \mathcal{R}$,

$$\Pr_S[h_S(x) = u,\ h_S(x') = v] = \frac{1}{M^2}.$$

Summing over $v$ shows that each $h_S(x)$ is uniform. The display then says that any two outputs on distinct fixed inputs are independent. It does *not* assert mutual independence of all outputs. We index by seeds so that even a family containing repeated functions has an unambiguous sampling rule. These conventions agree with [Vad12, Section 3.5.2].

## 6.2 Affine functions over a finite field

Let $q$ be a prime power. Choose $A, B$ independently and uniformly from $\mathbb{F}_q$, including $A = 0$, and set

$$h_{A,B}(x) = Ax + B \qquad (x \in \mathbb{F}_q).$$

**Claim.** This family is pairwise independent.

*Proof Sketch.* Fix $x \neq x'$ and desired outputs $u, v$. A seed $(a, b)$ produces these outputs precisely when

$$ax + b = u, \qquad ax' + b = v.$$

Subtracting determines the slope, and then the first equation determines the intercept:

$$a = \frac{u - v}{x - x'}, \qquad b = u - ax.$$

Division is valid because $x - x' \neq 0$ in a field. There is exactly one successful seed among $q^2$ equally likely seeds. Thus the probability is $q^{-2}$, as required. □

### 6.3 A mild relaxation: universal hashing

A family is *2-universal* if, for every fixed $x \neq x'$,

$$\Pr_S[h_S(x) = h_S(x')] \leq \frac{1}{M}.$$

Pairwise independence implies this condition: sum the probability $M^{-2}$ over the $M$ equal output pairs $(u, u)$. The converse is false. For example, the family $h_A(x) = Ax$ with $A$ uniform in $\mathbb{F}_q$ has collision probability exactly $1/q$ on distinct inputs, because a collision occurs exactly when $A = 0$. Yet $h_A(0) = 0$ always, so its outputs are not uniformly distributed at every input.

### 6.4 Application: bounded hash collisions

Fix distinct keys $x_1, \ldots, x_m$ *before* choosing the seed. If $C$ counts colliding unordered pairs under a pairwise-independent family with $M$ outputs, then

$$\mathbb{E}C = \sum_{i<j} \Pr[h_S(x_i) = h_S(x_j)] = \binom{m}{2}\frac{1}{M}.$$

For a merely 2-universal family, the equality becomes an upper bound. Linearity of expectation needs no independence between the collision events.

Pairwise independence also supports averaging. For a fixed function $g : \mathcal{R} \to [0, 1]$, let

$$Z_i = g(h_S(x_i)), \qquad \mu = \frac{1}{M}\sum_{y\in\mathcal{R}} g(y), \qquad \overline{Z} = \frac{1}{m}\sum_{i=1}^{m} Z_i.$$

Each $Z_i$ has mean $\mu$, and the $Z_i$ are pairwise independent because they are functions of pairwise-independent outputs. Since $0 \leq Z_i \leq 1$, we have $Z_i^2 \leq Z_i$, so $\mathbb{E}[Z_i^2] \leq \mu$ and hence $\mathrm{Var}(Z_i) \leq \mu - \mu^2 = \mu(1-\mu) \leq 1/4$. Therefore (applying early observations about pairwise independence and Chebyshev),

$$\mathbb{E}\overline{Z} = \mu, \quad \mathrm{Var}(\overline{Z}) \leq \frac{1}{4m}, \quad \Pr[|\overline{Z} - \mu| \geq \varepsilon] \leq \frac{1}{4m\varepsilon^2}.$$

We have recovered a useful sampling guarantee from a short seed. Technically, proof uses only one-variable means and two-variable products; it never inspects a joint distribution of three or more outputs. If you find yourself in such a situation, you can probably replace full joint independence with bounded independence [Vad12, Section 3.5.4].

**Warning**

The inputs in the definition are fixed independently of the seed. Choosing new inputs after learning the seed or earlier outputs does not follow. (Lookup "Collision-resistant hashing" or "adversarial robustness" to see learn about how to achieve restricted forms of these stronger guarantees.)

### 6.5 Other constructions (with essentially the same proof)

No need to remember all of this, but in case you are curious.

### Binary matrix maps

Choose an $m \times n$ matrix $A$ and a column vector $b \in \mathbb{F}_2^m$ with all entries independent and uniform. With $x$ regarded as a column, define

$$h_{A,b}(x) = Ax + b.$$

For a fixed nonzero $d = x - x'$, each row of $A$ has a uniform dot product with $d$, and the rows are independent. Thus $Ad$ is uniform in $\mathbb{F}_2^m$. Conditional on any $A$, the independent shift $b$ makes $Ax + b$ uniform. Consequently, for any prescribed $u, v$,

$$\Pr[Ax + b = u,\ Ax' + b = v] = \Pr[Ad = u - v]\, 2^{-m} = 2^{-2m}.$$

This family uses $mn + m$ random bits. Without the shift it remains 2-universal but has a fixed value at 0. This is the matrix version of the same distinction [Vad12, Problem 3.3].

### Shorter seeds by field multiplication

For $m \le n$, identify $\{0,1\}^n$ with $\mathbb{F}_{2^n}$ using a fixed basis over $\mathbb{F}_2$, and let $\pi_m$ select $m$ basis coordinates. Choose $a \in \mathbb{F}_{2^n}$ and $b \in \mathbb{F}_2^m$ uniformly and independently, and put

$$h_{a,b}(x) = \pi_m(ax) + b.$$

This uses $n + m$ bits. For fixed $x \neq x'$, multiplication by $x - x'$ permutes the field, so $a(x - x')$ is uniform. Its projection is uniform because every projected value has $2^{n-m}$ preimages. The independent shift then gives the same pair-probability calculation as above. Here multiplication is field multiplication, not integer multiplication modulo $2^n$ [Vad12, Theorem 3.26].

**A more general version of this (viewing the extension field as a vector space).** Restricting the binary matrix construction to matrices constant along each diagonal (*Toeplitz matrices*) gives another pairwise-independent family after adding the independent shift. Its seed has $n + 2m - 1$ bits. The proof is a separate linear-algebra exercise; see [Vad12, Problem 3.3(2)].

### Higher independence from interpolation

Choose $a_0, \dots, a_{k-1}$ independently and uniformly in $\mathbb{F}_q$, with $1 \le k \le q$, and let

$$h(x) = \sum_{j=0}^{k-1} a_j x^j.$$

For any $k$ distinct fixed inputs and any $k$ desired outputs, interpolation gives exactly one polynomial of degree less than $k$. Thus the probability of that output tuple is $q^{-k}$. Smaller tuples are uniform by marginalization, proving *$k$-wise independence*. The seed is $k$ field elements. This is the same evaluation map used in polynomial coding, now applied to *random* coefficients [Vad12, Section 3.5.5].

## 7 The mathematical language of theoretical computer science

### 7.1 Algorithms, randomness, and quantifiers

An algorithm is a procedure with a finite description. Its time and memory are measured as functions of the input length. A randomized algorithm is written $A(x; r)$: after fixing the input $x$ and the random tape $r$, the output is determined. Unless specified otherwise, $r$ is uniform among binary strings of the stated length and independent of the input. An output may therefore be random even for a fixed input.

## 7.2 Worst-case and average-case success

Let $f : \{0,1\}^n \to \{0,1\}$ be the function to be computed. The statement

$$\forall x \in \{0,1\}^n, \qquad \Pr_r[A(x;r) = f(x)] \geq \frac{2}{3}$$

is a *worst-case bounded-error guarantee*: every fixed input has success probability at least $2/3$ over the algorithm's coins. It does not say that one choice of coins works for all inputs.

Given an input distribution $D$, an *average-case guarantee* is instead

$$\Pr_{X \sim D,\, r}[A(X;r) = f(X)] \geq \frac{2}{3}.$$

It averages over both the input and the coins. A worst-case guarantee implies this for every $D$, by averaging the per-input inequalities. A guarantee for one fixed $D$ does not imply a worst-case guarantee.

Usually, we are interested in the case that $D$ is uniformly random, but not always (and in this case $D$ should be specified).

## 7.3 Averaging can fix coins, but not quantifiers

By iterated expectation,

$$\Pr_{X \sim D, r}[A(X;r) = f(X)] = \mathbb{E}_r\big[\Pr_{X \sim D}[A(X;r) = f(X)]\big].$$

If this average is at least $2/3$, some fixed tape $r^*$ gives average success at least $2/3$ over $D$. This is an existence statement, not an efficient way to find that tape. It does not imply the existence of a tape correct on every input. Problem 17 asks you to exhibit that distinction.

## 7.4 Circuits, truth tables, advice, and oracles

A Boolean function on $n$ bits has $2^n$ values in its truth table. A Boolean circuit computes a function through a finite directed acyclic graph of gates. Here use AND and OR gates with two inputs and NOT gates with one input; *size* counts non-input gates. The circuit can be far smaller than its truth table.

A *nonuniform circuit family* chooses a separate circuit $C_n$ for each input length, without requiring an algorithm that constructs $C_n$. Similarly, *advice* $a_n$ is an extra string allowed to depend on the input length but not on the particular input. Fixing a different successful tape $r_n^*$ at each length naturally gives such length-dependent information; it does not automatically give a uniform deterministic algorithm.

An algorithm with *oracle access* to $g$, written $A^g$, can query values $g(z)$ at inputs of its choice. One must separately specify the cost and number of queries. This notation will later describe algorithms that inspect only a few positions of a long word.

## 7.5 Computational Hardness

When we say a computational problem is *hard* we are claiming that there is no successful algorithm or circuit that solves it (or often, no *efficient* and successful algorithm or circuit).

A failure of one algorithm says nothing by itself about the hardness of (computing) $f$: another algorithm might work perfectly. To make a hardness claim, specify a class of allowed algorithms and show that *all* of them fail.

For a concrete finite model, let $\mathcal{C}_s$ be all deterministic circuits on $n$ inputs of size at most $s$. Worst-case hardness against this class means

$$\forall C \in \mathcal{C}_s \; \exists x \in \{0,1\}^n : \; C(x) \neq f(x).$$

This is pretty weak as each circuit might fail on just a single input. And that failed input may depend on $C$. An average-case hardness statement with error threshold $\delta > 0$ under $D$ is stronger:

$$\forall C \in \mathcal{C}_s : \quad \Pr_{X \sim D}[C(X) \neq f(X)] \geq \delta.$$

When $D = U_n$, this says every small circuit disagrees with the truth table on at least a $\delta$ fraction of its positions. For Boolean outputs, an upper bound of $1/2 + \varepsilon$ on success is equivalently a lower bound of $1/2 - \varepsilon$ on error. The constant $1/2$ is the success of an independent, arbitrary/random guess.

A worst-case-to-average-case *reduction* translates the first kind of hardness to the second: or in contrapostive form (how prove these things constructively), it turns a sufficiently accurate average-case solver for a transformed function into a solver for the original function on every input, with specified resource costs.

### 7.6 Communication protocols

In a two-party protocol, Alice receives $X$, Bob receives $Y$, and they exchange messages according to a protocol. The transcript, $\tau$, of such a protocol is simply the messages exchanged (in that order). A protocol is just a *next message function*: given my own internal state and the transcript thus far, what message should I send next? The full transcript $\tau$ is itself a random variable determined by the inputs and any public or private coins.

## 8 A non-exhaustive list of some helpful proof patterns

### 8.1 Indicators and linearity

To count a random collection, define one indicator for each candidate and sum. The expectation of this sum bypasses any dependence between the events.

### 8.2 Averaging and a good coordinate

If nonnegative numbers satisfy $\frac{1}{n}\sum_i a_i \leq \delta$ with $0 < \delta < 1$, then most coordinates cannot be much larger than $\delta$. For example, Markov gives

$$\left|\{i : a_i > \sqrt{\delta}\}\right| \leq \sqrt{\delta}\, n.$$

This is an example of what we often call an "averaging argument" (but this can also refer to the simpler fact that at least one coordinate must be at least as large as a lower bound on the average).

### 8.3 Condition, then account for the cost

Conditioning on an event can create correlations. A careful proof names the conditioning event, verifies it has positive probability, and tracks how the conditional distribution differs from the original distribution. (This is often helpful in hybrid arguments: see below).

## 8.4 Union bound and probabilistic existence

List all bad events, bound each one, sum the bounds, and conclude that some object avoids them all. The art lies in choosing the right random object and the right bad events.

## 8.5 Algebraic uniqueness

To show that two low-degree objects agree, subtract them and count roots. To prove that a linear object is uniquely determined, take a difference and place it in a kernel. These "difference plus uniqueness" arguments recur throughout algebraic constructions.

## 8.6 Encoding and incompressibility

To upper-bound the size of a family, show that every object has a uniquely decodable description from a limited set of bit strings. To prove that a bad event is rare or impossible, show that a bad object would admit an implausibly short description.

## 8.7 Telescoping and changing one component at a time

For any real numbers $a_0, \dots, a_m$,

$$a_m - a_0 = \sum_{i=1}^{m} (a_i - a_{i-1}), \qquad |a_m - a_0| \le \sum_{i=1}^{m} |a_i - a_{i-1}|.$$

If the left side is at least $\delta$, some step has size at least $\delta/m$. In a *hybrid argument*, $a_i$ is the probability of the same event under experiment $i$, and consecutive experiments differ in one component. This is an ordinary telescoping sum plus the triangle inequality, with an interpretation that will recur in the course [**?**, Appendix D.7].

## 8.8 Counting descriptions without losing a bit

There are $2^L$ binary strings of length exactly $L$, but $1 + 2 + \cdots + 2^L = 2^{L+1} - 1$ strings of length at most $L$. An injective encoding into the latter set does not by itself give the sharper bound $2^L$. Parsing restrictions can change the count: in a binary tree, a word of length $\ell \le L$ has $2^{L-\ell}$ descendants at depth $L$. This finite counting observation will explain prefix-free coding, without any need for an infinite probability space.

## 9 Some simple review problems

**Problem 1: Exponential bookkeeping**

Let $R, \alpha > 0$ be constants.

(a) Simplify $2^{nR}2^{-n\alpha}$.

(b) For which relation between $R$ and $\alpha$ does this quantity tend to zero?

(c) Suppose there are at most $2^{nR}$ bad events, each of probability at most $2^{-n\alpha}$. What does the union bound give?

**Problem 2: Asymptotic literacy**

Decide whether each assertion is true or false, and justify your answer.

(a) $n^{10} = 2^{o(n)}$.

(b) $1/n = 2^{-\Omega(n)}$.

(c) $n! = 2^{\Theta(n \log n)}$.

(d) If $\varepsilon_n \to 0$, then $n\varepsilon_n \to 0$.

**Problem 3: Jensen in numbers**

Let $Z$ equal $1/2$ and $1/8$, each with probability $1/2$.

(a) Compute $\mathbb{E}[-\log Z]$.

(b) Compute $-\log \mathbb{E}[Z]$.

(c) Verify Jensen's inequality and state when equality holds for a positive $Z$.

(d) For any nonnegative finite random variable $W$, use concavity to prove $\mathbb{E}[\sqrt{W}] \le \sqrt{\mathbb{E}W}$.

**Problem 4: A noisy bit**

Let $X \sim \mathrm{Bern}(1/2)$, let $N \sim \mathrm{Bern}(p)$ be independent, and set $Y = X \oplus N$.

(a) Show that $Y$ is uniform.

(b) Compute $\Pr[X = Y]$.

(c) Compute $\Pr[X = 0 \mid Y = 0]$.

(d) For which $p$ are $X$ and $Y$ independent?

**Problem 5: Pairwise does not mean mutual**

Let $X, Y$ be independent fair bits and $Z = X \oplus Y$.

(a) Show that every pair among $X, Y, Z$ is independent.

(b) Show that the triple is not mutually independent.

(c) Compute the conditional distribution of $(X, Y)$ given $Z = 0$. Are $X$ and $Y$ independent under this conditional distribution?

### Problem 6: A Markov chain by construction

Suppose $X \sim p_X$, then $Y \sim K(\cdot \mid X)$, and finally $Z \sim L(\cdot \mid Y)$.

(a) Write the joint distribution $p(x, y, z)$.

(b) Show that $p(z \mid x, y) = p(z \mid y)$ whenever $p_{X,Y}(x, y) > 0$.

(c) Interpret the conclusion as conditional independence.

(d) For a real-valued function $g$, express $\mathbb{E}[g(Z) \mid Y = y]$ using $L$, and obtain $\mathbb{E}[g(Z)]$ by averaging over $Y$.

### Problem 7: Collision counting

Let $X_1, \ldots, X_m$ be independent and uniform on $[N]$, and let $C$ count unordered pairs $i < j$ with $X_i = X_j$.

(a) Compute $\mathbb{E}C$.

(b) Bound the probability of at least one collision.

(c) If $m = \lfloor 2^{Rn} \rfloor$ and $N = 2^n$, for which constant $R > 0$ does your bound tend to zero?

(d) Let $X, X'$ instead be independent draws from an arbitrary distribution $P$ on $[N]$. Prove $\Pr[X = X'] = \sum_x P(x)^2 \geq 1/N$, and identify the equality case. Use Cauchy–Schwarz for the inequality.

### Problem 8: A weak distribution and its hypotheses

Let $X_1, \ldots, X_n$ be independent Bern($p$) variables and let $\overline{X} = n^{-1} \sum_i X_i$.

(a) Compute $\mathbb{E}\overline{X}$ and $\mathrm{Var}(\overline{X})$.

(b) Use Chebyshev to bound $\Pr[|\overline{X} - p| \geq \varepsilon]$.

(c) Explain why the bound tends to zero for every fixed $\varepsilon > 0$.

(d) Which steps still work with only pairwise independence? Does this also justify applying Hoeffding's inequality?

(e) Fix $0 < p < 1$. For a shrinking tolerance $\varepsilon_n = n^{-\alpha}$, $\alpha > 0$, when does the Chebyshev upper bound tend to zero? Explain the difference between a bound failing to vanish and a proof that the actual probability fails to vanish.

### Problem 9: Two averaging moves

(a) Let $a_1, \ldots, a_n \geq 0$ have average at most $\delta$, where $0 < \delta < 1$. Prove that at least $(1 - \sqrt{\delta})n$ indices satisfy $a_i \leq \sqrt{\delta}$.

(b) Let $b_0, \ldots, b_m \in [0, 1]$ satisfy $|b_m - b_0| \geq \eta > 0$. Prove that some $i \in [m]$ satisfies $|b_i - b_{i-1}| \geq \eta/m$.

### Problem 10: Counting error patterns

This is a counting review in Hamming-space language; see Section 4.4. Let $0 \le t \le n$ be an integer.

(a) Count binary strings of weight exactly $k$, and derive $V_2(n,t) = \sum_{j=0}^{t} \binom{n}{j}$.

(b) Reconstruct the proof that $V_2(n, \lfloor \rho n \rfloor) \le 2^{nh_2(\rho)}$ for $0 \le \rho \le 1/2$. Name the auxiliary probability distribution and the step that uses $\rho \le 1/2$.

(c) For fixed $0 < \rho < 1/2$, use the outermost layer and Stirling to find $\log V_2(n, \lfloor \rho n \rfloor)$ up to $O(\log n)$. What fraction of the cube is in this ball, at exponential scale?

(d) If distinct elements of $C \subseteq \{0,1\}^n$ have distance at least $2t+1$, prove that their radius-$t$ balls are disjoint and deduce $|C| V_2(n,t) \le 2^n$. Apply this with $n = 7$, $t = 1$ to bound $|C|$.

### Problem 11: Types

Let $\mathcal{X}$ be an alphabet of size $q$.

(a) Prove that length-$n$ strings have at most $(n+1)^q$ different types.

(b) If a type has symbol counts $n_1, \dots, n_q$, compute the size of its type class.

(c) If $X_1, \dots, X_n$ each have distribution $P$, prove $\mathbb{E}[\widehat{P}_{X^n}(a)] = P(a)$ for every symbol $a$. Is independence needed for this identity?

### Problem 12: The entropy-shaped binomial coefficient

Fix $p \in (0,1)$ and put $k = \lfloor pn \rfloor$. Use Stirling's approximation to prove

$$\log \binom{n}{k} = nh_2(p) + O(\log n).$$

First obtain the expression with $h_2(k/n)$, then justify replacing $k/n$ by $p$. The constant in the $O(\log n)$ term may depend on the fixed $p$.

### Problem 13: A kernel, a basis, and an error-detection preview

Work over $\mathbb{F}_2$. Set

$$H = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}, \qquad C = \{x \in \mathbb{F}_2^4 : Hx^\top = 0\}.$$

Use the linear-algebra discussion in Section 5.3; the term *codeword* just means a vector in $C$.

(a) Parametrize all vectors in $C$, and list them.

(b) Find $\dim C$ and $|C|$ using rank–nullity.

(c) Find a generator matrix $G$ and verify $HG^\top = 0$. Why is this identity alone not sufficient without the appropriate ranks?

(d) Find the minimum nonzero weight and explain why it equals the minimum distance between distinct vectors in $C$.

(e) Compute the syndromes of the four single-bit error vectors. Does the syndrome always detect a single-bit change? Does it always identify which bit changed? Exhibit any ambiguity.

### Problem 14: Polynomial uniqueness

Let $\mathbb{F}$ be a field.

(a) Prove that a nonzero polynomial of degree at most $d$ over $\mathbb{F}$ has at most $d$ roots.

(b) Deduce that two polynomials of degree less than $k$ that agree on $k$ distinct field elements must be identical.

### Problem 15: Interpolation over $\mathbb{F}_5$

Find the unique polynomial $p(x) \in \mathbb{F}_5[x]$ of degree less than two satisfying

$$p(1) = 2, \qquad p(3) = 4.$$

Then evaluate $p$ at $0, 1, 2, 3$.

### Problem 16: Using a pairwise-independent hash family

Read Section 6 first. Let $q$ be a prime power and choose $A, B$ independently and uniformly in $\mathbb{F}_q$.

(a) Fix $x \neq x'$ and prescribed outputs $u, v$. Solve for the unique seed $(a, b)$ with $ax + b = u$ and $ax' + b = v$, and deduce the output-pair probability.

(b) Compare two modifications: (i) set $B = 0$ and keep $A$ uniform in $\mathbb{F}_q$; (ii) keep $B$ uniform but require $A \neq 0$. For each, is the family 2-universal? Is it pairwise independent? Justify both answers.

(c) For an integer $r \geq 1$, now hash vectors $x \in \mathbb{F}_q^r$ to $\mathbb{F}_q$ by $h_{a,b}(x) = \sum_{j=1}^{r} a_j x_j + b$, with all $r+1$ seed entries independent and uniform. For fixed distinct $x, x'$, count the seeds giving each prescribed output pair. Deduce pairwise independence.

(d) In the scalar affine family, fix $1 \leq m \leq q$ distinct inputs $x_1, \ldots, x_m \in \mathbb{F}_q$ and a subset $T \subseteq \mathbb{F}_q$. Let $Z_i = \mathbf{1}_{\{h_{A,B}(x_i) \in T\}}$ and $p = |T|/q$. Compute the mean and variance of $m^{-1} \sum_i Z_i$, and give a Chebyshev bound for its deviation from $p$. All inputs and $T$ are fixed before the seed is sampled.

### Problem 17: Reading success and hardness statements

Use Section 7. Let $f : \{0,1\}^n \to \{0,1\}$.

(a) Translate $\forall x,\ \Pr_r[A(x; r) = f(x)] \geq 2/3$ and $\Pr_{X \sim U_n, r}[A(X; r) = f(X)] \geq 2/3$ into words. Prove the first implies the second; give a concrete counterexample to the converse.

(b) If the average success under an input distribution $D$ is at least $2/3$, prove that some fixed tape gives success at least $2/3$ over $D$. What does this say, and not say, about finding that tape?

(c) On the four inputs in $\{0,1\}^2$, construct a randomized algorithm for the constant-zero function that succeeds with probability at least $2/3$ on every input, but for which no fixed tape is correct on all four inputs.

(d) For a class $\mathcal{C}_s$ of circuits, compare $\forall C \in \mathcal{C}_s\ \exists x :\ C(x) \neq f(x)$ with $\forall C \in \mathcal{C}_s :\ \Pr_{X \sim D}[C(X) \neq f(X)] \geq \delta$. Which implies the other when $\delta > 0$? Why is showing that just one circuit fails not a hardness proof?

**Problem 18: Counting circuits**

Fix the gate set consisting of fan-in-two AND and OR and unary NOT. For $s \geq 1$, show that a circuit of size at most $s$ on $k$ inputs can be described in

$$O\big(s \log(s+k)\big)$$

bits, accounting also for the choice of output. Use this to prove that some Boolean function on $k$ bits requires circuit size $\Omega(2^k/k)$.

# 10 Quick reference

All logarithms below are base two; ln and $e^x$ use the natural base. All random variables have finite support. Denominators must be positive wherever a conditional probability is used.

## Probability and averaging

For $p_Y(y) > 0$,

$$p_{X|Y}(x \mid y) = \frac{p_{X,Y}(x,y)}{p_Y(y)}.$$

If $W$ is determined by $Y$,

$$\mathbb{E}[\mathbb{E}[Z \mid Y] \mid W] = \mathbb{E}[Z \mid W].$$

In particular, $\mathbb{E}[\mathbb{E}[Z \mid Y]] = \mathbb{E}Z$.

$$\Pr\left[\bigcup_i E_i\right] \le \sum_i \Pr[E_i].$$

For $Z \ge 0$ and $a > 0$,

$$\Pr[Z \ge a] \le \frac{\mathbb{E}Z}{a}.$$

For $t > 0$,

$$\Pr[|X - \mathbb{E}X| \ge t] \le \frac{\operatorname{Var}(X)}{t^2}.$$

Pairwise-independent real variables satisfy

$$\operatorname{Var}\left(\sum_i X_i\right) = \sum_i \operatorname{Var}(X_i).$$

For mutually independent $X_i \in [0,1]$,

$$\Pr[|\overline{X} - \mathbb{E}\overline{X}| \ge \varepsilon] \le 2e^{-2n\varepsilon^2} \quad (\varepsilon > 0).$$

Here $\overline{X} = n^{-1}\sum_i X_i$. Chebyshev for a mean with variance $\sigma^2/n$ gives $\sigma^2/(n\varepsilon_n^2)$; it vanishes when $n\varepsilon_n^2 \to \infty$.

## Convexity and norms

For convex $\varphi$,

$$\varphi(\mathbb{E}X) \le \mathbb{E}[\varphi(X)].$$

For a concave function the direction reverses. Strict convexity gives equality only for a constant random variable, ignoring zero-probability values.

$$\left(\sum_i a_i b_i\right)^2 \le \left(\sum_i a_i^2\right)\left(\sum_i b_i^2\right).$$

$$\left(\sum_{i=1}^{M} |a_i|\right)^2 \le M \sum_{i=1}^{M} a_i^2.$$

$$\ln u \le u - 1 \quad (u > 0), \qquad 1 - v \le e^{-v} \quad (v \in \mathbb{R}).$$

## Counting and Hamming balls

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}, \qquad \binom{n}{n_1, \ldots, n_q} = \frac{n!}{\prod_i n_i!}.$$

Stirling at logarithmic scale:

$$\log(n!) = n\log n - (\log e)n + O(\log n).$$

With $0 \log 0 = 0$,

$$h_2(p) = -p\log p - (1-p)\log(1-p).$$

For fixed $0 < p < 1$,

$$\log \binom{n}{\lfloor pn \rfloor} = nh_2(p) + O(\log n).$$

For integer $0 \le t \le n$,

$$V_2(n,t) = \sum_{j=0}^{t} \binom{n}{j}.$$

For $0 \le \rho \le 1/2$,

$$V_2(n, \lfloor \rho n \rfloor) \le 2^{nh_2(\rho)}.$$

Distance at least $2t+1$ gives

$$|C| V_2(n,t) \le 2^n.$$

## Algebra and hashing

For $A : \mathbb{F}_q^n \to \mathbb{F}_q^r$ of rank $s$,

$$|\ker A| = q^{n-s}.$$

Every nonempty fiber is a kernel translate. A $k$-dimensional code has $q^k$ vectors; full-rank generators and parity checks satisfy $HG^\top = 0$.

Pairwise independence, for fixed $x \ne x'$ and $M$ output values, means

$$\Pr_S[h_S(x) = u, h_S(x') = v] = M^{-2}.$$

The weaker 2-universal condition is

$$\Pr_S[h_S(x) = h_S(x')] \le M^{-1}.$$

Uniform $A, B \in \mathbb{F}_q$ give $h_{A,B}(x) = Ax + B$. Include $A = 0$. A random polynomial of degree less than $k$ gives $k$-wise independence at distinct field points, for $k \le q$.

## References and further reading

[Vad12] Salil P. Vadhan. *Pseudorandomness*. *Foundations and Trends in Theoretical Computer Science*, 7(1–3):1–336, 2012. The most direct companion to this tutorial is Section 3.5: Section 3.5.2 and Construction 3.23 give affine hashing; Section 3.5.4 explains pairwise-independent sampling; Section 3.5.5 gives higher independence from polynomials. Problem 3.3 treats binary matrices and Toeplitz matrices. Sections 7.4–7.5 give later context for average-case hardness and coding-based reductions.

[CW79] J. Lawrence Carter and Mark N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18(2):143–154, 1979. The original universal-hashing paper; read it for the motivation behind controlling collisions without choosing a fully random function table.

[WC81] Mark N. Wegman and J. Lawrence Carter. New hash functions and their use in authentication and set equality. *Journal of Computer and System Sciences*, 22(3):265–279, 1981. Further hashing constructions and applications. The probability condition used by a construction should always be checked explicitly; "universal" and "pairwise independent" are not interchangeable here.

[GRS25] Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. *Essential Coding Theory*. Draft dated August 26, 2025. Chapter 2 develops linear codes. Section 3.3.1 studies Hamming-ball volumes and their entropy exponents.

For a broader collection of standard inequalities, I have found László Kozma's *Useful Inequalities Cheat Sheet* invaluable.