# Application: Perfect hashing

Handling collisions via "**two-level hashing**"

- First level hash table has size O(N)
- Each location in the hash table performs a collision-free hashing

Let C(i) = number of elements mapped to location i in the first level table

Q: For the second level table, what should the table size at location i?

C(i)^2 (We know that for this size, we can find a collision-free hash function)

# Application: Perfect hashing

Q: What is the total table space used in the second level?

$$\sum_{i=1}^{M} C(i)^2$$

We know $E(C) = \binom{N}{2}\frac{1}{M} \Rightarrow E\left[\sum_{i=1}^{M}\binom{C(i)}{2}\right] = \binom{N}{2}\frac{1}{M}$

$$E\left[\sum_{i=1}^{M} C(i)^2 - \sum_{i=1}^{M} C(i)\right] = O(N) \quad \text{since } M = O(N)$$

$$\Rightarrow E\left[\sum_{i=1}^{M} C(i)^2\right] = O(N) \quad \text{since } E\left[\sum_{i=1}^{M} C(i)\right] = O(N)$$

Q: What is the total table space?

O(N)

Collision-free and O(N) table space!

# k-wise independent hash functions

In addition to universality, certain independence properties of hash functions are useful in analysis of algorithms

**Definition.** A family H of hash functions mapping U to [M] is called k-wise-independent if for any k distinct keys

$x_1, x_2, \ldots x_k$ and any k distinct values $\alpha_1, \alpha_2, \ldots, \alpha_k$

we have

$$P(h(x_1) = \alpha_1 \cap h(x_2) = \alpha_2 \cap \ldots \cap h(x_k) = \alpha_k) \leq \frac{1}{m^k}$$

Case for k=2 is called “pairwise independent.

# k-wise independent hash functions

**Properties:**

Suppose H is a k-wise independent family for k>=2. Then

1. H is also (k-1)-wise indepdent.
2. For any $x \in U$ and $a \in [M]$ $P[h(x) = a] <= 1/M$.
3. H is universal.

Q: Which is stronger: pairwise independent or universal?

Pairwise independent is stronger.

E.g.?

$h(x) = Ax$ construction since $P[h(0) = 0] = 1$

# Some constructions: 2-wise independent

Construction 1 (variant of random matrix multiplication):

Let A be a m x u matrix with uniformly random binary entries.

Let b be a m-bit vector with uniformly random binary entries.

$$h(x) := Ax + b$$

where the arithmetic is modulo 2.

**Claim.** This family of hash functions is 2-wise independent.

Q: How many hash functions are in this family?

$2^{(u+1)m}$

Q: Number of bits to store?

O(um)

Can we do with fewer bits?

# Some constructions: 2-wise independent

Construction 2 (Using fewer bits):

Let A be a m x u matrix.

- Fill the first row and column with uniformly random binary entries.
- Set $A_{i,j} = A_{i-1,j-1}$

Let b be a m-bit vector with uniformly random binary entries.

$$h(x) := Ax + b$$

where the arithmetic is modulo 2.

**Claim.** This family of hash functions is 2-wise independent.

(try to proof this yourself)

# Some constructions: 2-wise independent

Construction 3 (**Using finite fields**)

**Switch to slides for a primer on Groups, fields and finite fields**

We will need this again when we learn about algorithms for coding.

So we will digress a bit to learn/recap about these number theory basics.

# Groups

A **Group** (G,*,I) is a set *G* with operator * such that:

1. **Closure.** For all $a,b \in G$, $a * b \in G$
2. **Associativity.** For all $a,b,c \in G$, $a*(b*c) = (a*b)*c$
3. **Identity.** There exists $I \in G$, such that for all $a \in G$, $a*I=I*a=a$
4. **Inverse.** For every $a \in G$, there exist a unique element $b \in G$, such that $a*b=b*a=I$

An **Abelian or Commutative Group** is a Group with the additional condition

5. **Commutativity.** For all $a,b \in G$, $a*b=b*a$

# Examples of groups

Q: Examples?

- Integers, Reals or Rationals with Addition
- The nonzero Reals or Rationals with Multiplication
- Non-singular n x n real matrices with
  Matrix Multiplication
- Permutations over n elements with composition
  $[0\to1, 1\to2, 2\to0] \circ [0\to1, 1\to0, 2\to2] = [0\to0, 1\to2, 2\to1]$

Often we will be concerned with **finite groups**, I.e., ones with a finite number of elements.

# Groups based on modular arithmetic

The group of positive integers modulo a prime *p*

$Z_p^* \equiv \{1, 2, 3, \ldots, p\text{-}1\}$ $*_p \equiv$ multiplication modulo p

Denoted as: $(Z_p^*, *_p)$

**Required properties**

1. Closure. Yes.
2. Associativity. Yes.
3. Identity. 1.
4. Inverse. Yes. (try to prove this yourself)

**Example:** $Z_7^* = \{1,2,3,4,5,6\}$

$1^{-1} = 1$, $2^{-1} = 4$, $3^{-1} = 5$, $6^{-1} = 6$

# Fields

A **Field** is a set of elements F with **two** binary operators * and + such that

1. (F, +) is an **abelian group**
2. (F \ $I_+$, *) is an **abelian group**
   the "multiplicative group"
3. **Distribution**: $a*(b+c) = a*b + a*c$
4. **Cancellation**: $a*I_+ = I_+$

Example: The reals and rationals with + and * are fields.

The **order (or size)** of a field is the number of elements.

A field of finite order is a **finite field**.

# Finite Fields

$\mathbb{Z}_p$ (p prime) with + and * mod p, is a **finite** field.

1. ($\mathbb{Z}_p$, +) is an **abelian group** (0 is identity)
2. ($\mathbb{Z}_p \setminus 0$, $*$) is an **abelian group** (1 is identity)
3. **Distribution**: a*(b+c) = a*b + a*c
4. **Cancellation**: a*0 = 0

We denote this by $\mathbb{F}_p$ or GF(p)

Are there other finite fields?

What about ones that fit nicely into bits, bytes and words (i.e with $2^k$ elements)?

# Polynomials over $\mathbb{F}_p$

$\mathbb{F}_p[x]$ = polynomials on x with coefficients in $\mathbb{F}_p$.

- Example of $\mathbb{F}_5[x]$: f(x) = $3x^4 + 1x^3 + 4x^2 + 3$
- deg(f(x)) = 4 (the **degree** of the polynomial)

Operations: (examples over $\mathbb{F}_5[x]$)

•Addition: $(x^3 + 4x^2 + 3) + (3x^2 + 1) = (x^3 + 2x^2 + 4)$

•Multiplication: $(x^3 + 3) * (3x^2 + 1) = 3x^5 + x^3 + 4x^2 + 3$

•$I_+ = 0$, $I_* = 1$

•+ and * are associative and commutative

•Multiplication distributes and 0 cancels

Do these polynomials form a field?

# Division and Modulus

Long division on polynomials ($\mathbb{F}_5[x]$):

$$
\begin{array}{r|l}
 & \boxed{1x + 4} \\
x^2 + 1 & x^3 + 4x^2 + 0x + 3 \\
 & \underline{x^3 + 0x^2 + 1x + 0} \\
 & 4x^2 + 4x + 3 \\
 & \underline{4x^2 + 0x + 4} \\
 & \boxed{4x + 4}
\end{array}
$$

$$(x^3 + 4x^2 + 3)/(x^2 + 1) = (x + 4)$$

$$(x^3 + 4x^2 + 3) \bmod (x^2 + 1) = (4x + 4)$$

$$(x^2 + 1)(x + 4) + (4x + 4) = (x^3 + 4x^2 + 3)$$

# Polynomials modulo Polynomials

How about making a field of polynomials modulo another polynomial?

This is analogous to $\mathbb{F}_p$ (i.e., integers modulo another integer).

Need a polynomial analogous to a prime number…

**Definition:** An **irreducible polynomial** is one that is not a product of two other polynomials both of degree greater than 0.

e.g. $(x^2 + 2)$ for $\mathbb{F}_5[x]$

# Galois Fields

The polynomials $\mathbb{F}_p[x] \bmod p(x)$ where

*1.* $p(x) \in$ ∈ $\mathbb{F}_p$ $[x]$, p(x) is irreducible and

2. deg(p(x)) = n

form a finite field.

Q: How many elements?

Such a field has $p^n$ elements.

These fields are called **Galois Fields** or **GF($p^n$)** or $\mathbb{F}_{p^n}$

The special case n = 1 reduces to the fields $\mathbb{F}_p$.

The special case p = 2 is especially useful for us.

# GF($2^n$)

$\mathbb{F}_{2^n}$ = set of polynomials in $\mathbb{F}_2[x]$ modulo irreducible polynomial $\mathrm{p}(x) \in \mathbb{F}_2[x]$ of degree $n$.

Elements are all polynomials in $\mathbb{F}_2[x]$ of degree $\leq n - 1$.
Has $2^n$ elements.
Natural correspondence with bits in $\{0,1\}^n$.

Elements of $\mathbb{F}_{2^8}$ can be represented as **a byte**, one bit for each term.

*E.g.*, $x^6 + x^4 + x + 1 = 01010011$

# GF($2^n$)

$\mathbb{F}_{2^n}$ = set of polynomials in $\mathbb{F}_2[x]$ modulo irreducible polynomial $\mathrm{p}(x) \in \mathbb{F}_2[x]$ of degree $n$.

Elements are all polynomials in $\mathbb{F}_2[x]$ of degree $\leq n-1$.

Has $2^n$ elements.

Natural correspondence with bits in $\{0,1\}^n$.

**Addition** over $\mathbb{F}_2$ corresponds to xor.

- Just take the xor of the bit-strings (bytes or words in practice). This is dirt cheap.

# Multiplication over GF($2^n$)

If n is small enough can use a table of all combinations.

The size will be $2^n$ x $2^n$ (e.g. 64K for $\mathbb{F}_{2^8}$)

Otherwise, use standard shift and add (xor)

**Note**: dividing through by the irreducible polynomial on an overflow by 1 term is simply a test and an xor.

e.g. 0111 mod 1001 = 0111

1011 mod 1001 = 1011 xor 1001 = 0010

^ just look at this bit for $\mathbb{F}_{2^3}$

# Finding inverses over $GF(2^n)$

Again, if n is small just store in a table.

- Table size is just $2^n$.

For larger n, use Euclid's algorithm.

- This is again easy to do with shift and xors.

# Euclid's Algorithm

**Euclid's Algorithm**:

gcd(a,b) = gcd(b,a mod b)

gcd(a,0) = a

**"Extended" Euclid's algorithm**:

- Find **x** and **y** such that **ax + by = gcd(a,b)**
- Can be calculated as a side-effect of Euclid's algorithm.
- Note that **x** and **y** can be zero or negative.

This allows us to find **$a^{-1}$ mod p**, for **a** $\in Z_p^*$

Q: Any idea how?

In particular return **x** in **ax + py = 1**.

Similarly can apply to over polynomials