Universal and Perfect Hashing

Should tables be sorted?

universal hashing

now: hash function chosen randomly

def: *universal families of hash functions*

- Let HF be a family of functions

$$h : U \rightarrow [0 : m-1] \text{ for } h \in HF$$

- Draw random $h \in HF$ with uniform distribution

$$W = (HF, p)\ ,\ p(h) = \frac{1}{\#HF}$$

- HF is called *universal* if for all $x, y \in U$

$$\boxed{x \neq y \ \rightarrow\ p\{h \mid h(x) = h(y)\} \leq 1/m}$$

example. not practical

$$HP = \{h \mid h : U \rightarrow [0 : m-1]\}$$

universal hashing

now: hash function chosen randomly

def: *universal families of hash functions*

- Let HF be a family of functions

$$h : U \to [0 : m-1] \text{ for } h \in HF$$

- Draw random $h \in HF$ with uniform distribution

$$W = (HF, p)\ ,\ p(h) = \frac{1}{\#HF}$$

- HF is called *universal* if for all $x, y \in U$

$$\boxed{x \neq y \ \to\ p\{h \mid h(x) = h(y)\} \leq 1/m}$$

example. not practical

$$HP = \{h \mid h : U \to [0 : m-1]\}$$

Lemma 4. *With randomly drawn hash function from universal family of hash functions expected list lenghth is* $\leq 1 + \frac{n-1}{m}$.

universal hashing

now: hash function chosen randomly

def: *universal families of hash functions*

- Let HF be a family of functions

$$h : U \to [0 : m-1] \text{ for } h \in HF$$

- Draw random $h \in HF$ with uniform distribution

$$W = (HF, p)\ ,\ p(h) = \frac{1}{\#HF}$$

- HF is called *universal* if for all $x, y \in U$

$$\boxed{x \neq y \ \to \ p\{h \mid h(x) = h(y)\} \leq 1/m}$$

example. not practical

$$HP = \{h \mid h : U \to [0 : m-1]\}$$

Lemma 4. *With randomly drawn hash function from universal family of hash functions expected list lenghth is* $\leq 1 + \frac{n-1}{m}$.

- let $x \in U$. List length for x is random variable

$$X(h) = \#\{k \in S \mid h(k) = h(x)\}$$

universal hashing

now: hash function chosen randomly

def: *universal families of hash functions*

- Let HF be a family of functions

$$h : U \rightarrow [0 : m-1] \text{ for } h \in HF$$

- Draw random $h \in HF$ with uniform distribution

$$W = (HF, p)\ ,\ p(h) = \frac{1}{\#HF}$$

- HF is called *universal* if for all $x, y \in U$

$$\boxed{x \neq y \ \rightarrow \ p\{h \mid h(x) = h(y)\} \leq 1/m}$$

example. not practical

$$HP = \{h \mid h : U \rightarrow [0 : m-1]\}$$

Lemma 4. *With randomly drawn hash function from universal family of hash functions expected list lenghth is* $\leq 1 + \frac{n-1}{m}$.

- let $x \in U$. List length for x is random variable

$$X(h) = \#\{k \in S \mid h(k) = h(x)\}$$

- indicator variables for $k \in S$

$$Y_k(h) = \begin{cases} 1 & h(k) = h(x) \\ 0 & \text{otherwise} \end{cases}$$

$$X(h) = \sum_{k \in S} Y_k(h)$$

universal hashing

now: hash function chosen randomly

def: *universal families of hash functions*

- Let HF be a family of functions

$$h : U \to [0 : m-1] \text{ for } h \in HF$$

- Draw random $h \in HF$ with uniform distribution

$$W = (HF, p)\ ,\ p(h) = \frac{1}{\#HF}$$

- HF is called *universal* if for all $x, y \in U$

$$\boxed{x \neq y \ \to\ p\{h \mid h(x) = h(y)\} \leq 1/m}$$

example. not practical

$$HP = \{h \mid h : U \to [0 : m-1]\}$$

Lemma 4. *With randomly drawn hash function from universal family of hash functions expected list lenghth is* $\leq 1 + \frac{n-1}{m}$.

- let $x \in U$. List length for x is random variable

$$X(h) = \#\{k \in S \mid h(k) = h(x)\}$$

- indicator variables for $k \in S$

$$Y_k(h) = \begin{cases} 1 & h(k) = h(x) \\ 0 & \text{otherwise} \end{cases}$$

$$X(h) = \sum_{k \in S} Y_k(h)$$

-

$$\begin{aligned} E[Y_k] &= p\{h \in HF \mid Y_k(h) = 1\} \text{ (indicator variable)} \\ &= p\{h \in HF \mid h(k) = h(x)\} \\ & \begin{cases} \leq 1/m & k \neq x \\ = 1 & k = x \end{cases} \quad (HF \text{ universal}) \end{aligned}$$

universal hashing

now: hash function chosen randomly

def: *universal families of hash functions*

- Let HF be a family of functions

$$h : U \rightarrow [0 : m-1] \text{ for } h \in HF$$

- Draw random $h \in HF$ with uniform distribution

$$W = (HF, p) \ , \ p(h) = \frac{1}{\#HF}$$

- HF is called *universal* if for all $x, y \in U$

$$\boxed{x \neq y \ \rightarrow \ p\{h \mid h(x) = h(y)\} \leq 1/m}$$

example. not practical

$$HP = \{h \mid h : U \rightarrow [0 : m-1]\}$$

Lemma 4. *With randomly drawn hash function from universal family of hash functions expected list lenghth is* $\leq 1 + \frac{n-1}{m}$.

- let $x \in U$. List length for x is random variable

$$X(h) = \#\{k \in S \mid h(k) = h(x)\}$$

- indicator variables for $k \in S$

$$Y_k(h) = \begin{cases} 1 & h(k) = h(x) \\ 0 & \text{otherwise} \end{cases}$$

$$X(h) = \sum_{k \in S} Y_k(h)$$

-

$$\begin{aligned} E[Y_k] &= p\{h \in HF \mid Y_k(h) = 1\} \text{ (indicator variable)} \\ &= p\{h \in HF \mid h(k) = h(x)\} \\ & \begin{cases} \leq 1/m & k \neq x \\ = 1 & k = x \end{cases} \quad (HF \text{ universal}) \end{aligned}$$

-

$$\begin{aligned} E[X] &= E[\sum_{k \in S} Y_k(h)] \\ &= \sum_{k \in S} E[Y_k] \quad \text{(linearity)} \\ &= E[Y_x] + \sum_{k \in S \setminus \{x\}} E[Y_k] \\ &\leq 1 + \frac{n-1}{m} \end{aligned}$$

linear hashing

a practical universal family of hash functions

$$R = (\mathbb{B}, + \bmod 2, \cdot \bmod 2, 0, 1)$$

$$\bmod 2 = \oplus \;, \; \cdot \bmod 2 = \wedge$$

R is a ring. Proof: exercise.

- $$U = \mathbb{B}^\nu \;, \; m = 2^\mu$$
- code hash values $h(x) \in [0 : 2^\mu - 1]$ as binary numbers in $\mathbb{B}^\mu$.
- family of hash functions

$$H_{lin} = \{h : \mathbb{B}^\nu \to \mathbb{B}^\mu \mid h \text{ linear}\}$$

- hash function $h \in H_{lin}$ specified by matrix $M_h \in \mathbb{B}^{\mu \times \nu}$.
- choose bits of M independently each with probability $1/2$

$$p(M) = 2^{-\mu \cdot \nu}$$

linear hashing

a practical universal family of hash functions

$$R = (\mathbb{B}, + \bmod 2, \cdot \bmod 2, 0, 1)$$

$$\bmod 2 = \oplus \;,\; \cdot \bmod 2 = \wedge$$

R is a ring. Proof: exercise.

- $$U = \mathbb{B}^\nu \;,\; m = 2^\mu$$
- code hash values $h(x) \in [0 : 2^\mu - 1]$ as binary numbers in $\mathbb{B}^\mu$.
- family of hash functions

$$H_{lin} = \{h : \mathbb{B}^\nu \to \mathbb{B}^\mu \mid h \text{ linear}\}$$

- hash function $h \in H_{lin}$ specified by matrix $M_h \in \mathbb{B}^{\mu \times \nu}$.
- choose bits of M independently each with probability $1/2$

$$p(M) = 2^{-\mu \cdot \nu}$$

computing a hash value

- for $x \in \mathbb{B}^\nu$ computing $h(x)$ is a matrix-vector product

$$h(x) = Mx$$

linear hashing

a practical universal family of hash functions

$$R = (\mathbb{B}, + \bmod 2, \cdot \bmod 2, 0, 1)$$

$$\bmod 2 = \oplus \ , \ \cdot \bmod 2 = \wedge$$

R is a ring. Proof: exercise.

- $$U = \mathbb{B}^{\nu} \ , \ m = 2^{\mu}$$
- code hash values $h(x) \in [0 : 2^{\mu} - 1]$ as binary numbers in $\mathbb{B}^{\mu}$.
- family of hash functions

$$H_{lin} = \{h : \mathbb{B}^{\nu} \to \mathbb{B}^{\mu} \mid h \text{ linear}\}$$

- hash function $h \in H_{lin}$ specified by matrix $M_h \in \mathbb{B}^{\mu \times \nu}$.
- choose bits of M independently each with probability $1/2$

$$p(M) = 2^{-\mu \cdot \nu}$$

computing a hash value

- for $x \in \mathbb{B}^{\nu}$ computing $h(x)$ is a matrix-vector product

$$\begin{pmatrix} M_{1,1} & M_{1,i} & M_{1,\nu} \\ \vdots & & \\ M_{j,1} & M_{j,i} & M_{j,\nu} \\ M_{\mu,1} & M_{\mu,i} & M_{\mu,\nu} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_j \\ x_i \\ \vdots \\ x_\nu \end{pmatrix}$$

$$h(x) = Mx$$

$$h(x)_j = \sum_{i=1}^{\nu} M_{j,i} \cdot x_i$$

linear hashing

a practical universal family of hash functions

$$R = (\mathbb{B}, + \bmod 2, \cdot \bmod 2, 0, 1)$$

$$\bmod 2 = \oplus \ , \ \cdot \bmod 2 = \wedge$$

R is a ring. Proof: exercise.

-
$$U = \mathbb{B}^\nu \ , \ m = 2^\mu$$

- code hash values $h(x) \in [0 : 2^\mu - 1]$ as binary numbers in $\mathbb{B}^\mu$.
- family of hash functions

$$H_{lin} = \{h : \mathbb{B}^\nu \to \mathbb{B}^\mu \mid h \text{ linear}\}$$

- hash function $h \in H_{lin}$ specified by matrix $M_h \in \mathbb{B}^{\mu \times \nu}$.
- choose bits of M independently each with probability $1/2$

$$p(M) = 2^{-\mu \cdot \nu}$$

computing a hash value

- for $x \in \mathbb{B}^\nu$ computing $h(x)$ is a matrix-vector product

$$\begin{pmatrix} M_{1,1} & M_{1,i} & M_{1,\nu} \\ \vdots & \vdots & \\ M_{j,1} & M_{j,i} & M_{j,\nu} \\ M_{\mu,1} & M_{\mu,i} & M_{\mu,\nu} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_j \\ x_i \\ \vdots \\ x_\nu \end{pmatrix}$$

$$h(x) = Mx$$

$$h(x)_j = \sum_{i=1}^{\nu} M_{j,i} \cdot x_i$$

- column vectors c_i of M:

$$c_i = \begin{pmatrix} M_{1,i} \\ \vdots \\ M_{\mu,i} \end{pmatrix}$$

$$M = (c_1, \ldots, c_\nu)$$

linear hashing

a practical universal family of hash functions

$$R = (\mathbb{B}, + \bmod 2, \cdot \bmod 2, 0, 1)$$

$$\bmod 2 = \oplus \;, \;\cdot \bmod 2 = \wedge$$

R is a ring. Proof: exercise.

- $$U = \mathbb{B}^\nu \;,\; m = 2^\mu$$
- code hash values $h(x) \in [0 : 2^\mu - 1]$ as binary numbers in $\mathbb{B}^\mu$.
- family of hash functions

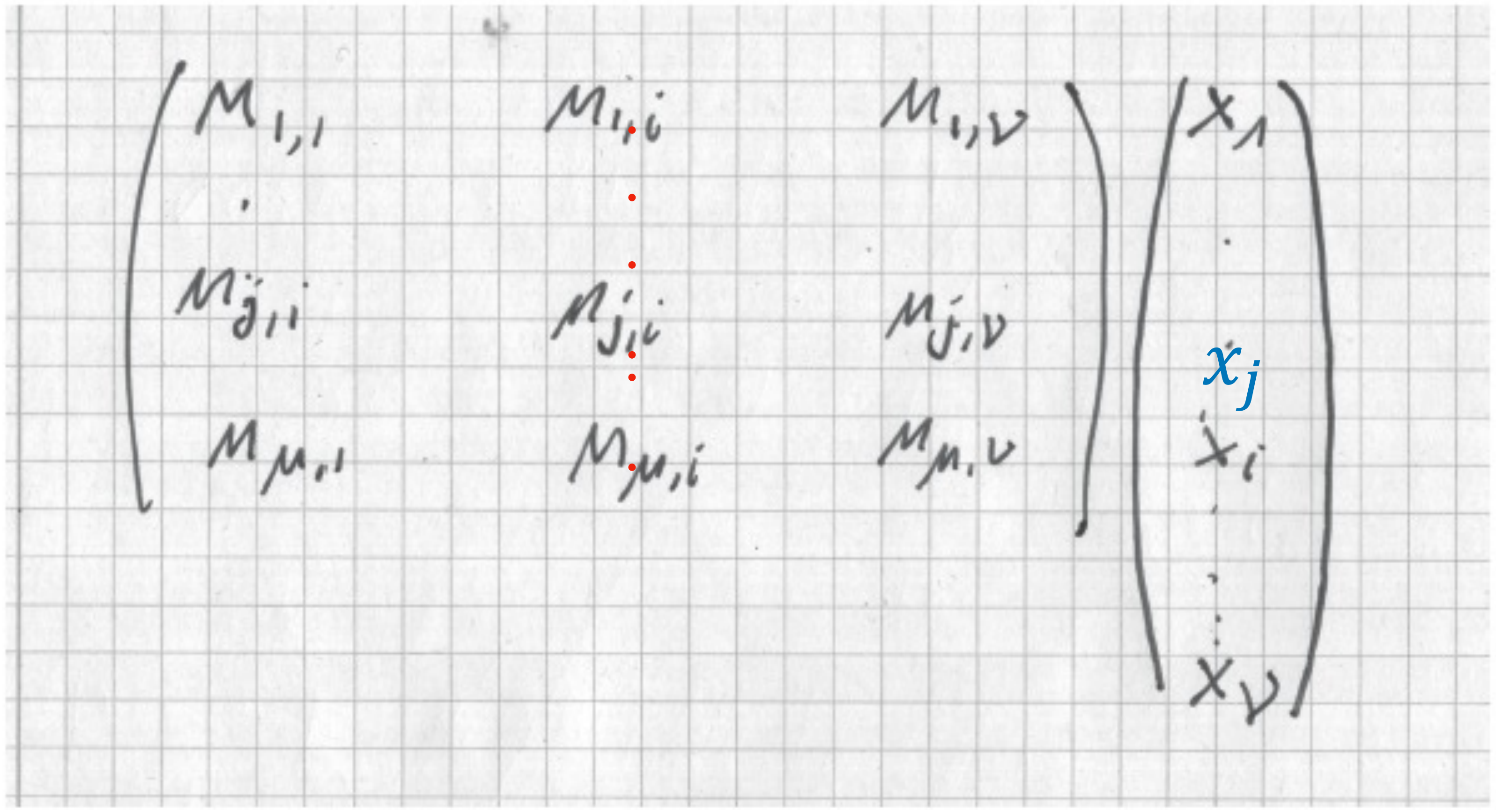
$$H_{lin} = \{h : \mathbb{B}^\nu \to \mathbb{B}^\mu \mid h \text{ linear}\}$$

- hash function $h \in H_{lin}$ specified by matrix $M_h \in \mathbb{B}^{\mu \times \nu}$.
- choose bits of M independently each with probability $1/2$

$$p(M) = 2^{-\mu \cdot \nu}$$

computing a hash value

- for $x \in \mathbb{B}^\nu$ computing $h(x)$ is a matrix-vector product



$$h(x) = Mx$$

$$h(x)_j = \sum_{i=1}^{\nu} M_{j,i} \cdot x_i$$

- column vectors c_i of M:

$$c_i = \begin{pmatrix} M_{1,i} \\ \vdots \\ M_{\mu,i} \end{pmatrix}$$

$$M = (c_1, \ldots, c_\nu)$$

$$c_i x_i = \begin{pmatrix} M_{1,i} \cdot x_i \\ \vdots \\ M_{\mu,i} \cdot x_i \end{pmatrix}$$

linear hashing

a practical universal family of hash functions

$$R = (\mathbb{B}, + \bmod 2, \cdot \bmod 2, 0, 1)$$

$$\bmod 2 = \oplus \ , \ \cdot \bmod 2 = \wedge$$

R is a ring. Proof: exercise.

- $$U = \mathbb{B}^\nu \ , \ m = 2^\mu$$
- code hash values $h(x) \in [0 : 2^\mu - 1]$ as binary numbers in $\mathbb{B}^\mu$.
- family of hash functions

$$H_{lin} = \{h : \mathbb{B}^\nu \to \mathbb{B}^\mu \mid h \text{ linear}\}$$

- hash function $h \in H_{lin}$ specified by matrix $M_h \in \mathbb{B}^{\mu \times \nu}$.
- choose bits of M independently each with probability $1/2$

$$p(M) = 2^{-\mu \cdot \nu}$$

computing a hash value

- for $x \in \mathbb{B}^\nu$ computing $h(x)$ is a matrix-vector product

$$\begin{pmatrix} M_{1,1} & M_{1,i} & M_{1,\nu} \\ \cdot & \vdots & \\ M_{j,1} & M_{j,i} & M_{j,\nu} \\ M_{\mu,1} & M_{\mu,i} & M_{\mu,\nu} \end{pmatrix} \begin{pmatrix} x_1 \\ \cdot \\ x_j \\ x_i \\ \vdots \\ x_\nu \end{pmatrix}$$

$$h(x) = Mx$$

$$h(x)_j = \sum_{i=1}^{\nu} M_{j,i} \cdot x_i$$

- column vectors c_i of M:

$$c_i = \begin{pmatrix} M_{1,i} \\ \vdots \\ M_{\mu,i} \end{pmatrix}$$

$$M = (c_1, \ldots, c_\nu)$$

$$c_i x_i = \begin{pmatrix} M_{1,i} \cdot x_i \\ \vdots \\ M_{\mu,i} \cdot x_i \end{pmatrix}$$

$$\boxed{\begin{aligned} \sum_{i=1}^{\nu} c_i x_i &= \begin{pmatrix} \sum_{i=1}^{\nu} M_{1,i} \cdot x_i \\ \vdots \\ \sum_{i=1}^{\nu} M_{\mu,i} \cdot x_i \end{pmatrix} \\ &= h(x) \end{aligned}}$$

universality of linear hashing

$$R = (\mathbb{B}, + \bmod 2, \cdot \bmod 2, 0, 1)$$

$$\bmod 2 = \oplus \;,\; \cdot \bmod 2 = \wedge$$

R is a ring. Proof: exercise.

- $$U = \mathbb{B}^\nu \;,\; m = 2^\mu$$
- code hash values $h(x) \in [0 : 2^\mu - 1]$ as binary numbers in $\mathbb{B}^\mu$.
- family of hash functions

$$H_{lin} = \{h : \mathbb{B}^\nu \to \mathbb{B}^\mu \mid h \text{ linear}\}$$

- hash function $h \in H_{lin}$ specified by matrix $M_h \in \mathbb{B}^{\mu \times \nu}$.
- choose bits of M independently each with probability $1/2$

$$p(M) = 2^{-\mu \cdot \nu}$$

$$\boxed{\begin{aligned} \sum_{i=1}^{\nu} c_i x_i &= \begin{pmatrix} \sum_{i=1}^{\nu} M_{1,i} \cdot x_i \\ \vdots \\ \sum_{i=1}^{\nu} M_{\mu,i} \cdot x_i \end{pmatrix} \\ &= h(x) \end{aligned}}$$

Lemma 5. *H_{lin} is universal*

- Let

$$x[1:\nu], y[1:\nu] \in \mathbb{B}^\nu \text{ with } \cancel{Mx \neq My}$$

$x \neq y, Mx = My$

- w.l.o.g. (without loss of generality)

$$x_1 \neq y_1 \;,\; x_1 = 0 \;,\; y_1 = 1$$

- c_i column vectors of M

$$M = (c_1, \ldots, c_\nu)$$

- $$h(x) = Mx = \sum_{i=1}^{\nu} c_i x_i = \sum_{i=2}^{\nu} c_i x_i$$

$$h(y) = Mx = \sum_{i=1}^{\nu} c_i y_i$$

$$h(x) = h(y) \;\leftrightarrow\; c_1 = \sum_{i=2}^{\nu} c_i (x_i - y_i)$$

$$R = (\mathbb{B}, + \bmod 2, \cdot \bmod 2, 0, 1)$$

$$\bmod 2 = \oplus \;, \; \cdot \bmod 2 = \wedge$$

R is a ring. Proof: exercise.

- $$U = \mathbb{B}^\nu \;, \; m = 2^\mu$$
- code hash values $h(x) \in [0 : 2^\mu - 1]$ as binary numbers in $\mathbb{B}^\mu$.
- family of hash functions

$$H_{lin} = \{h : \mathbb{B}^\nu \to \mathbb{B}^\mu \mid h \text{ linear}\}$$

- hash function $h \in H_{lin}$ specified by matrix $M_h \in \mathbb{B}^{\mu \times \nu}$.
- choose bits of M independently each with probability $1/2$

$$p(M) = 2^{-\mu \cdot \nu}$$

$$\boxed{\begin{aligned} \sum_{i=1}^{\nu} c_i x_i &= \begin{pmatrix} \sum_{i=1}^{\nu} M_{1,i} \cdot x_i \\ \vdots \\ \sum_{i=1}^{\nu} M_{\mu,i} \cdot x_i \end{pmatrix} \\ &= h(x) \end{aligned}}$$

Lemma 5. *H_{lin} is universal*

universality of linear hashing

- Let

$$x[1:\nu], y[1:\nu] \in \mathbb{B}^\nu \text{ with } \cancel{Mx \neq My}$$

$x \neq y, Mx = My$

- w.l.o.g. (without loss of generality)

$$x_1 \neq y_1 \;, \; x_1 = 0 \;, \; y_1 = 1$$

- c_i column vectors of M

$$M = (c_1, \ldots, c_\nu)$$

- $$h(x) = Mx = \sum_{i=1}^{\nu} c_i x_i = \sum_{i=2}^{\nu} c_i x_i$$

$$h(y) = Mx = \sum_{i=1}^{\nu} c_i y_i$$

$$h(x) = h(y) \;\leftrightarrow\; c_1 = \sum_{i=2}^{\nu} c_i (x_i - y_i)$$

- $$c_1 \text{ determined by } c_2, \ldots, c_\nu$$

$$R = (\mathbb{B}, + \bmod 2, \cdot \bmod 2, 0, 1)$$

$$\bmod 2 = \oplus \;,\; \cdot \bmod 2 = \wedge$$

R is a ring. Proof: exercise.

- $$U = \mathbb{B}^\nu \;,\; m = 2^\mu$$
- code hash values $h(x) \in [0 : 2^\mu - 1]$ as binary numbers in $\mathbb{B}^\mu$.
- family of hash functions

$$H_{lin} = \{h : \mathbb{B}^\nu \to \mathbb{B}^\mu \mid h \text{ linear}\}$$

- hash function $h \in H_{lin}$ specified by matrix $M_h \in \mathbb{B}^{\mu \times \nu}$.
- choose bits of M independently each with probability $1/2$

$$p(M) = 2^{-\mu \cdot \nu}$$

$$\boxed{\begin{aligned} \sum_{i=1}^{\nu} c_i x_i &= \begin{pmatrix} \sum_{i=1}^{\nu} M_{1,i} \cdot x_i \\ \vdots \\ \sum_{i=1}^{\nu} M_{\mu,i} \cdot x_i \end{pmatrix} \\ &= h(x) \end{aligned}}$$

Lemma 5. *H_{lin} is universal*

- Let

$$x[1:\nu], y[1:\nu] \in \mathbb{B}^\nu \text{ with } \cancel{Mx \neq My}$$

$x \neq y, Mx = My$

- w.l.o.g. (without loss of generality)

$$x_1 \neq y_1 \;,\; x_1 = 0 \;,\; y_1 = 1$$

- c_i column vectors of M

$$M = (c_1, \ldots, c_\nu)$$

- $$h(x) = Mx = \sum_{i=1}^{\nu} c_i x_i = \sum_{i=2}^{\nu} c_i x_i$$

$$h(y) = Mx = \sum_{i=1}^{\nu} c_i y_i$$

$$h(x) = h(y) \;\leftrightarrow\; c_1 = \sum_{i=2}^{\nu} c_i (x_i - y_i)$$

- c_1 determined by $c_2, \ldots, c_\nu$

- number of matrices satisfying this:

$$K = 2^{(\nu-1)\cdot\mu} = \#H_{lin}/2^\mu = \#H_{lin}/m$$

- $$p\{h \mid h(x) = h(y)\} = K/\#H_{lin} = 1/m$$

excursion 1:
throwing a coin until head comes up

$$S = \{0^i 1 \mid i \in \mathbb{N}_0\} \quad , \quad p(0^i 1) = \frac{1}{2^{i+1}}$$

Lemma 6. *(S, p) is a probability space*

$$\begin{aligned} \sum_{i=0}^{\infty} p(0^i 1) &= \frac{1}{2} \cdot \sum_{i=1}^{\infty} \frac{1}{2^i} \\ &= \frac{1}{2} \cdot 2 \end{aligned}$$

excursion 1:
throwing a coin until head comes up

$$S = \{0^i 1 \mid i \in \mathbb{N}_0\} \quad , \quad p(0^i 1) = \frac{1}{2^{i+1}}$$

Lemma 6. *(S, p) is a probability space*

$$\begin{aligned} \sum_{i=0}^{\infty} p(0^i 1) &= \frac{1}{2} \cdot \sum_{i=1}^{\infty} \frac{1}{2^i} \\ &= \frac{1}{2} \cdot 2 \end{aligned}$$

- number of throws: random variable

$$X(0^i 1) = i + 1$$

Lemma 7.

$$E(X) = O(1)$$

excursion 1:
throwing a coin until head comes up

$$S = \{0^i1 \mid i \in \mathbb{N}_0\} \quad , \quad p(0^i1) = \frac{1}{2^{i+1}}$$

Lemma 6. *(S,p) is a probability space*

$$\begin{aligned}\sum_{i=0}^{\infty} p(0^i1) &= \frac{1}{2} \cdot \sum_{i=1}^{\infty} \frac{1}{2^i} \\ &= \frac{1}{2} \cdot 2\end{aligned}$$

- number of throws: random variable

$$X(0^i1) = i+1$$

Lemma 7.

$$E(X) = O(1)$$

excursion 1:
throwing a coin until head comes up

$$S = \{0^i 1 \mid i \in \mathbb{N}_0\} \quad , \quad p(0^i 1) = \frac{1}{2^{i+1}}$$

Lemma 6. (S, p) *is a probability space*

$$\begin{aligned} \sum_{i=0}^{\infty} p(0^i 1) &= \frac{1}{2} \cdot \sum_{i=1}^{\infty} \frac{1}{2^i} \\ &= \frac{1}{2} \cdot 2 \end{aligned}$$

- number of throws: random variable

$$X(0^i 1) = i + 1$$

Lemma 7.

$$E(X) = O(1)$$

$$\begin{aligned} E(x) &= \sum_{i=0}^{\infty} \frac{i+1}{2^{i+1}} \\ &= \sum_{i=1}^{\infty} \frac{i}{2^i} \\ &= O(1) \end{aligned}$$

excursion 2:
Markov's inequality

Values of a nonnegative random variable X much above expected value are unlikely

Lemma 8. *Let X be a nonnegative random variable and $t > 0$. Then*

$$p\{s \mid X(s) \geq t\} \leq E[X]/t$$

excursion 2:
Markov's inequality

Values of a nonnegative random variable X much above expected value are unlikely

Lemma 8. *Let X be a nonnegative random variable and $t > 0$. Then*

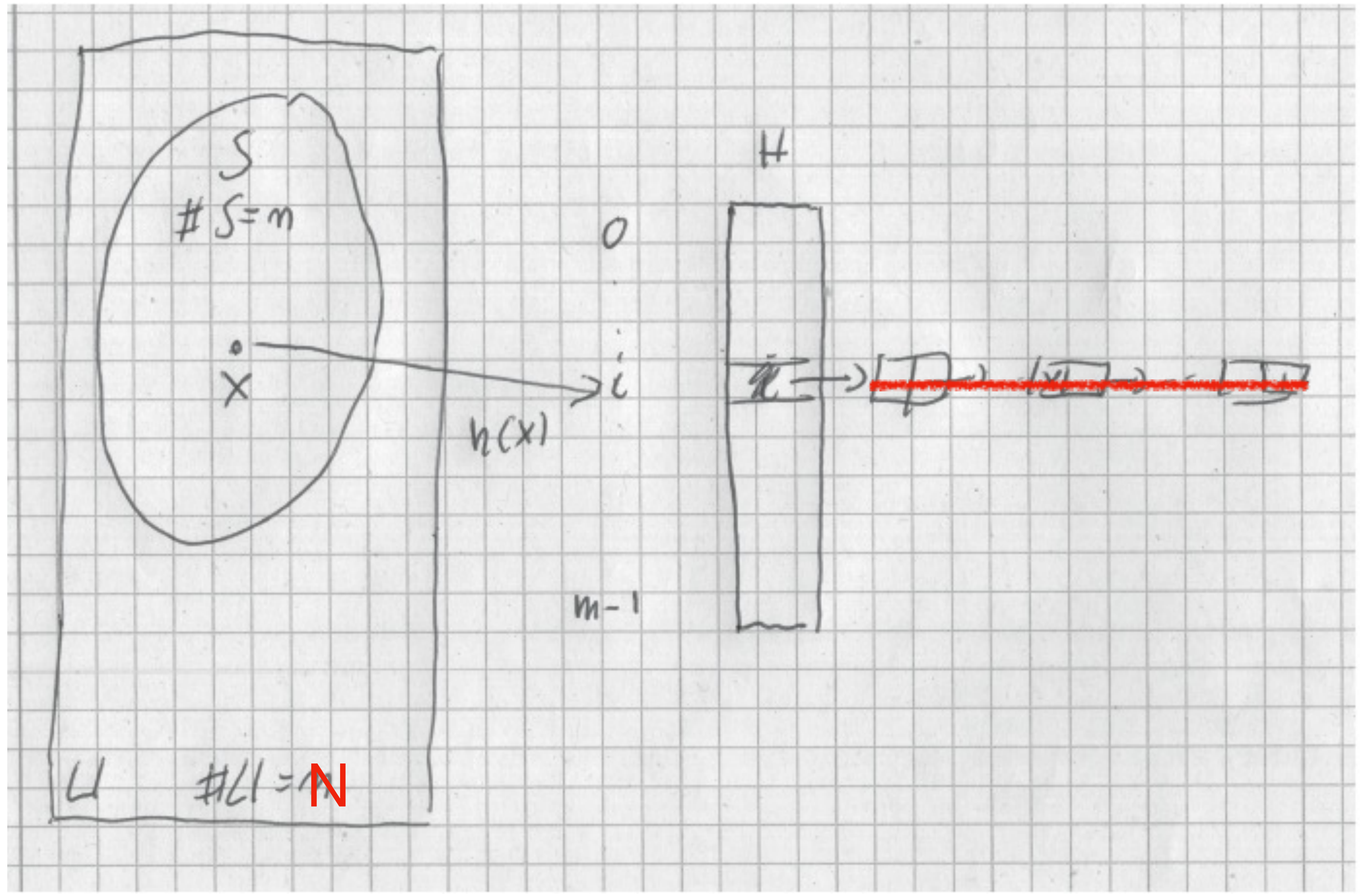
$$p\{s \mid X(s) \geq t\} \leq E[X]/t$$

$$\begin{aligned}
E[X] &= \sum_{s \in S} X(s) \cdot p(s) \\
&\geq \sum_{X(s) \geq t} X(s) \cdot p(s) \\
&\geq \sum_{X(s) \geq t} t \cdot p(s) \\
&= t \cdot \sum_{X(s) \geq t} p(s) \\
&= t \cdot p\{s \mid X(s) \geq t\}
\end{aligned}$$

perfect hashing with quadratic space

Lemma 9. *Let HF be a universal family of hash functions and $m = n^2$. Then*

$$p\{h \mid \forall (x \neq y)\; h(x) \neq h(y)\} \geq 1/2$$

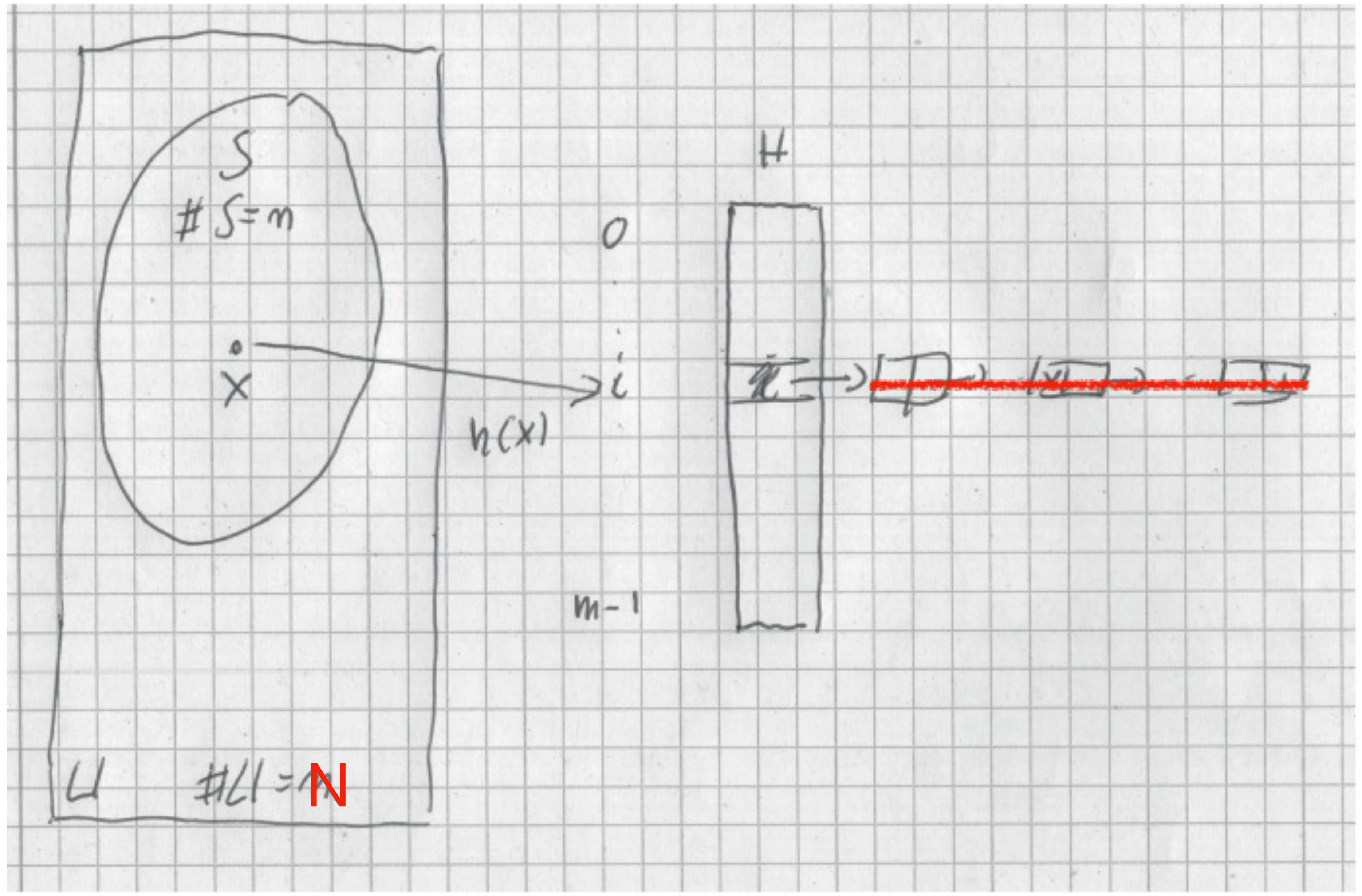


perfect hashing with quadratic space

Lemma 9. *Let HF be a universal family of hash functions and $m = n^2$. Then*

$$p\{h \mid \forall x \neq y.\ h(x) \neq h(y)\} \geq 1/2$$

$$\begin{aligned}
p\{h \mid \forall (x \neq y).\ h(x) \neq h(y)\} &= 1 - p\{h \mid \exists (x \neq y).\ h(x) = h(y)\} \\
&\geq 1 - \sum_{\{x,y\}\ ,\ x \neq y} p\{h \mid h(x) = h(y)\} \\
&\geq 1 - \binom{n}{2} \cdot \frac{1}{m} \quad (HF \text{ universal}) \\
&= 1 - \frac{n \cdot (n-1)}{2m} \\
&\geq 1 - \frac{n^2}{2m} \\
&\geq 1 - \frac{1}{2} \quad (m = n^2)
\end{aligned}$$

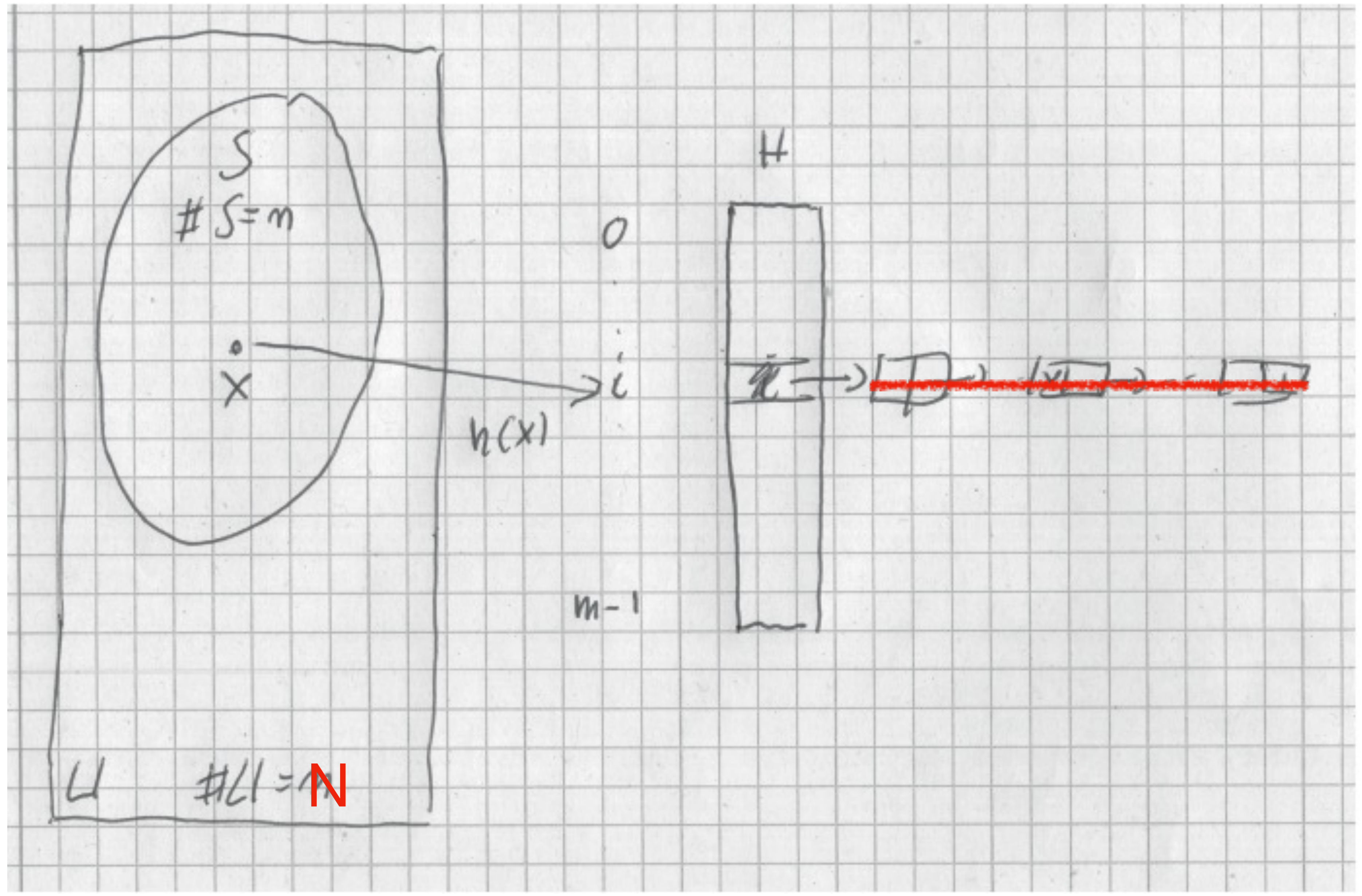


perfect hashing with quadratic space

Lemma 9. *Let HF be a universal family of hash functions and $m = n^2$. Then*

$$p\{h \mid \forall x \neq y.\ h(x) \neq h(y)\} \geq 1/2$$

$$
\begin{aligned}
p\{h \mid \forall(x \neq y).\ h(x) \neq h(y)\} &= 1 - p\{h \mid \exists(x \neq y).\ h(x) = h(y)\} \\
&\geq 1 - \sum_{\{x,y\}\ ,\ x \neq y} p\{h \mid h(x) = h(y)\} \\
&\geq 1 - \binom{n}{2} \cdot \frac{1}{m} \quad (HF \text{ universal}) \\
&= 1 - \frac{n \cdot (n-1)}{2m} \\
&\geq 1 - \frac{n^2}{2m} \\
&\geq 1 - \frac{1}{2} \quad (m = n^2)
\end{aligned}
$$



Lemma 7 $\rightarrow$ expected number of tries until such a function is picked is $O(1)$

perfect hashing with linear space

two stages of hashing

- primary stage

$$h : U \to [0 : n-1] \text{ from universal family } HF$$

- set of elements $x \in S$ mapped to $i \in [0 : n-1]$

$$S_i(h) = \{x \in S \mid h(x) = i\}$$

- size of the sets

$$m_i(h) = \#S_i(h)$$

- secondary hashing: map each $S_i(h)$ perfectly (without collisions) with lemma 9 by hash function

$$h_i : S_i(h) \to [0 : m_i(h)^2 - 1]$$

perfect hashing with linear space

two stages of hashing

- primary stage

$$h : U \to [0 : n-1] \text{ from universal family } HF$$

- set of elements $x \in S$ mapped to $i \in [0 : n-1]$

$$S_i(h) = \{x \in S \mid h(x) = i\}$$

- size of the sets

$$m_i(h) = \#S_i(h)$$

- secondary hashing: map each $S_i(h)$ perfectly (without collisions) with lemma 9 by hash function

$$h_i : S_i(h) \to [0 : m_i(h)^2 - 1]$$

- expected total length of secondary hash tables is linear

Lemma 10.

$$E[\sum_{i=0}^{n-1} m_i^2] < 2n.$$

perfect hashing with linear space

two stages of hashing

- primary stage

$$h : U \to [0 : n-1] \text{ from universal family } HF$$

- set of elements $x \in S$ mapped to $i \in [0 : n-1]$

$$S_i(h) = \{x \in S \mid h(x) = i\}$$

- size of the sets

$$m_i(h) = \#S_i(h)$$

- secondary hashing: map each $S_i(h)$ perfectly (without collisions) with lemma 9 by hash function

$$h_i : S_i(h) \to [0 : m_i(h)^2 - 1]$$

- expected total length of secondary hash tables is linear

Lemma 10.

$$E[\sum_{i=0}^{n-1} m_i^2] < 2n.$$

For $x, y \in S$ define

$$Z_{x,y}(h) = \begin{cases} 1 & h(x) = h(y) \\ 0 & \text{otherwise} \end{cases} \quad \text{(indicator variable)}$$

$$\sum_{i=0}^{n-1} m_i^2 = \sum_{(x,y)\in S\times S} Z_{x,y}$$

because for all i each pair $(x, y) \in S_i \times S_i$ contributes 1 to right hand side. Other pairs do not contribute.

perfect hashing with linear space

two stages of hashing

- primary stage

$$h : U \to [0 : n-1] \text{ from universal family } HF$$

- set of elements $x \in S$ mapped to $i \in [0 : n-1]$

$$S_i(h) = \{x \in S \mid h(x) = i\}$$

- size of the sets

$$m_i(h) = \#S_i(h)$$

- secondary hashing: map each $S_i(h)$ perfectly (without collisions) with lemma 9 by hash function

$$h_i : S_i(h) \to [0 : m_i(h)^2 - 1]$$

- expected total length of secondary hash tables is linear

Lemma 10.

$$E[\sum_{i=0}^{n-1} m_i^2] < 2n.$$

For $x, y \in S$ define

$$Z_{x,y}(h) = \begin{cases} 1 & h(x) = h(y) \\ 0 & \text{otherwise} \end{cases} \quad \text{(indicator variable)}$$

$$\sum_{i=0}^{n-1} m_i^2 = \sum_{(x,y)\in S\times S} Z_{x,y}$$

because for all i each pair $(x,y) \in S_i \times S_i$ contributes 1 to right hand side. Other pairs do not contribute.

$$\begin{aligned}
E[\sum_{i=0}^{n-1} m_i(h)^2] &= \sum_{x\in S}\sum_{y\in S} E[Z_{x,y}] \quad \text{(linearity)} \\
&= n + \sum_{x\in S}\sum_{y\neq x} E[Z_{x,y}] \\
&= n + \sum_{x\in S}\sum_{y\neq x} p\{h \mid h(x) = h(y)\} \quad \text{(indicator variable)} \\
&\leq n + \sum_{x\in S}\sum_{y\neq x} \frac{1}{n} \quad (HF \text{ universal}) \\
&= n + n(n-1)\cdot\frac{1}{n} \\
&< 2n
\end{aligned}$$

perfect hashing with linear space

two stages of hashing

- primary stage

$$h : U \to [0 : n-1] \text{ from universal family } HF$$

- set of elements $x \in S$ mapped to $i \in [0 : n-1]$

$$S_i(h) = \{x \in S \mid h(x) = i\}$$

- size of the sets

$$m_i(h) = \#S_i(h)$$

- secondary hashing: map each $S_i(h)$ perfectly (without collisions) with lemma 9 by hash function

$$h_i : S_i(h) \to [0 : m_i(h)^2 - 1]$$

- expected total length of secondary hash tables is linear

Lemma 10.

$$E[\sum_{i=0}^{n-1} m_i^2] < 2n.$$

For $x, y \in S$ define

$$Z_{x,y}(h) = \begin{cases} 1 & h(x) = h(y) \\ 0 & \text{otherwise} \end{cases} \quad \text{(indicator variable)}$$

$$\sum_{i=0}^{n-1} m_i^2 = \sum_{(x,y) \in S \times S} Z_{x,y}$$

because for all i each pair $(x, y) \in S_i \times S_i$ contributes 1 to right hand side. Other pairs do not contribute.

$$\begin{aligned} E[\sum_{i=0}^{n-1} m_i(h)^2] &= \sum_{x \in S} \sum_{y \in S} E[Z_{x,y}] \quad \text{(linearity)} \\ &= n + \sum_{x \in S} \sum_{y \neq x} E[Z_{x,y}] \\ &= n + \sum_{x \in S} \sum_{y \neq x} p\{h \mid h(x) = h(y)\} \quad \text{(indicator variable)} \\ &\leq n + \sum_{x \in S} \sum_{y \neq x} \frac{1}{n} \quad (HF \text{ universal}) \\ &= n + n(n-1) \cdot \frac{1}{n} \\ &< 2n \end{aligned}$$

- likelihood of finding h with short secondary tables: Lemma 10 and 8 (Markov's inequality) with $\cancel{t = 2.}$ $t = 4n$

$$p\{h \mid \sum_{i=0}^{n-1} m_i(h)^2 > 4n\} \leq 1/2$$

perfect hashing with linear space

two stages of hashing

- primary stage

$$h : U \to [0 : n-1] \text{ from universal family } HF$$

- set of elements $x \in S$ mapped to $i \in [0 : n-1]$

$$S_i(h) = \{x \in S \mid h(x) = i\}$$

- size of the sets

$$m_i(h) = \#S_i(h)$$

- secondary hashing: map each $S_i(h)$ perfectly (without collisions) with lemma 9 by hash function

$$h_i : S_i(h) \to [0 : m_i(h)^2 - 1]$$

- expected total length of secondary hash tables is linear

Lemma 10.

$$E[\sum_{i=0}^{n-1} m_i^2] < 2n.$$

For $x, y \in S$ define

$$Z_{x,y}(h) = \begin{cases} 1 & h(x) = h(y) \\ 0 & \text{otherwise} \end{cases} \quad \text{(indicator variable)}$$

$$\sum_{i=0}^{n-1} m_i^2 = \sum_{(x,y) \in S \times S} Z_{x,y}$$

because for all i each pair $(x,y) \in S_i \times S_i$ contributes 1 to right hand side. Other pairs do not contribute.

$$\begin{aligned}
E[\sum_{i=0}^{n-1} m_i(h)^2] &= \sum_{x \in S} \sum_{y \in S} E[Z_{x,y}] \quad \text{(linearity)} \\
&= n + \sum_{x \in S} \sum_{y \neq x} E[Z_{x,y}] \\
&= n + \sum_{x \in S} \sum_{y \neq x} p\{h \mid h(x) = h(y)\} \quad \text{(indicator variable)} \\
&\leq n + \sum_{x \in S} \sum_{y \neq x} \frac{1}{n} \quad (HF \text{ universal}) \\
&= n + n(n-1) \cdot \frac{1}{n} \\
&< 2n
\end{aligned}$$

- likelihood of finding h with short secondary tables: Lemma 10 and 8 (Markov's inequality) with ~~$t = 2$.~~ $t = 4n$

$$p\{h \mid \sum_{i=0}^{n-1} m_i(h)^2 > 4n\} \leq 1/2$$

Lemma 7 $\to$ expected number tries to find such h is $O(1)$.

possible implementations

two stages of hashing

- primary stage

$$h : U \to [0 : n-1] \text{ from universal family } HF$$

- set of elements $x \in S$ mapped to $i \in [0 : n-1]$

$$S_i(h) = \{x \in S \mid h(x) = i\}$$

- size of the sets

$$m_i(h) = \#S_i(h)$$

- secondary hashing: map each $S_i(h)$ perfectly (without collisions) with lemma 9 by hash function

$$h_i : S_i(h) \to [0 : m_i(h)^2 - 1]$$

```
i = h(x) /*primary hash value*/
```

- primary array H has pointers to secondary arrays H_i Look up x at

```
H[i]*[h_i(x)]
```

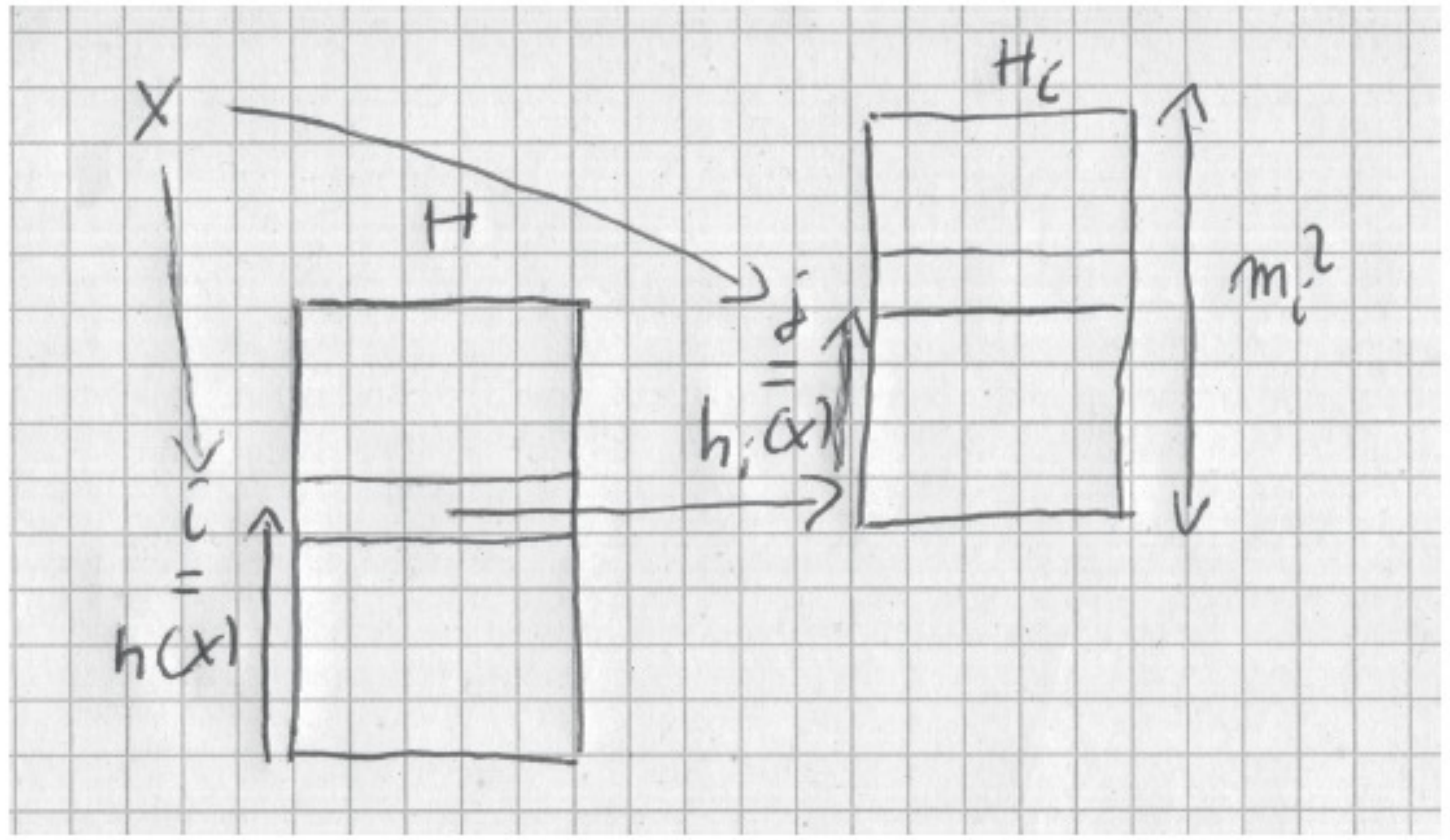


Figure 4: For $h(x) = i$ element $H[i]$ points to secondary array H_i. If present, x can be found at element $j = h_i(x)$ of that array

possible implementations

two stages of hashing

- primary stage

$$h : U \to [0 : n-1] \text{ from universal family } HF$$

- set of elements $x \in S$ mapped to $i \in [0 : n-1]$

$$S_i(h) = \{x \in S \mid h(x) = i\}$$

- size of the sets

$$m_i(h) = \#S_i(h)$$

- secondary hashing: map each $S_i(h)$ perfectly (without collisions) with lemma 9 by hash function

$$h_i : S_i(h) \to [0 : m_i(h)^2 - 1]$$

```
i = h(x) /*primary hash value*/
```

- concatenate secondary arrays to a single array $H2$. H_i starts at (precomputed) index

$$B[i] = \sum_{j<i} m_j^2$$

Look up x at

```
H2[B[i] + h_i(x)]
```

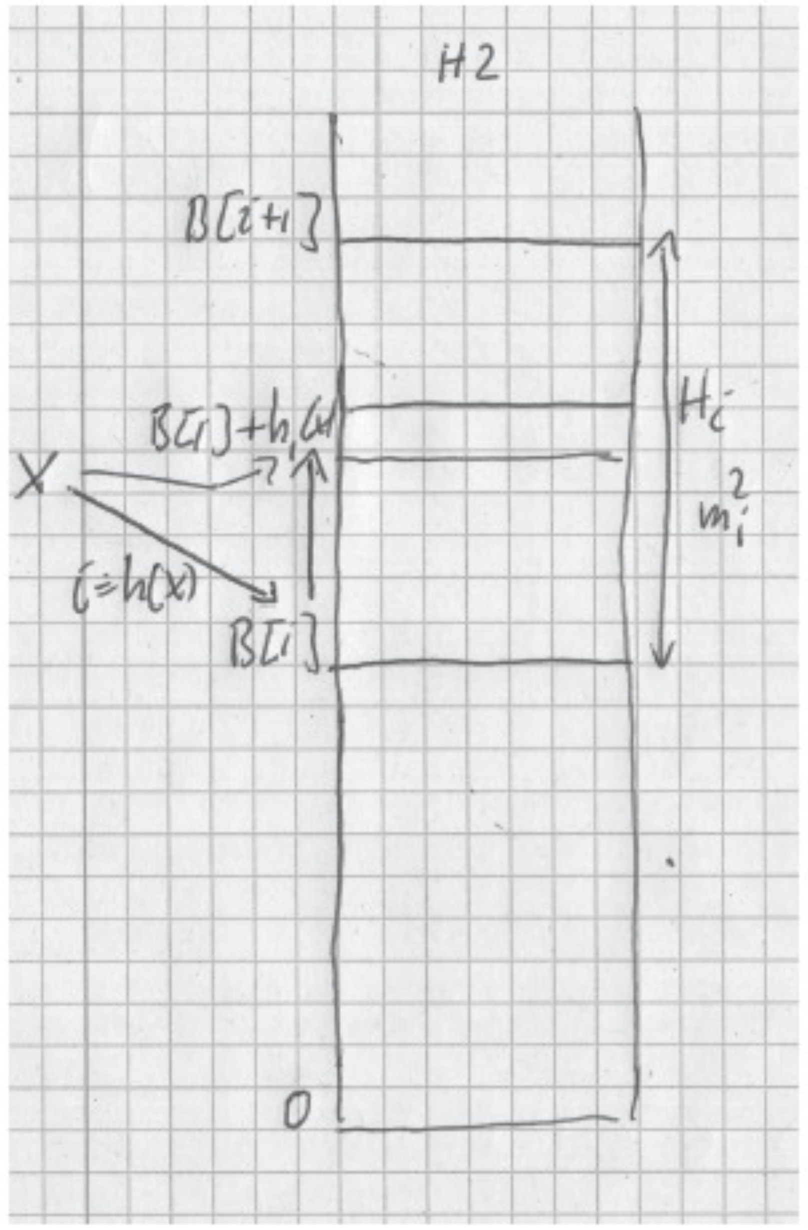


Figure 5: For $h(x) = i$ the base address of portion H_i in $H2$ is computed as $B[i] = \Sigma_{j<i} m_j^2$. If present x can be found at element $H2[B[i] + j]$ where $j = h_i(x)$.