

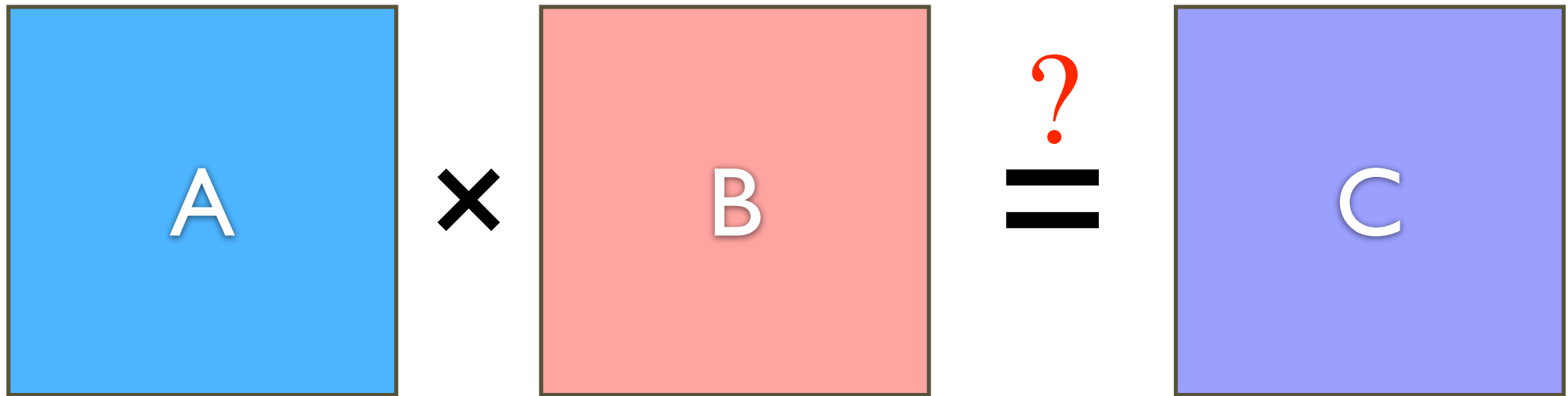
Advanced Algorithms

Fingerprinting

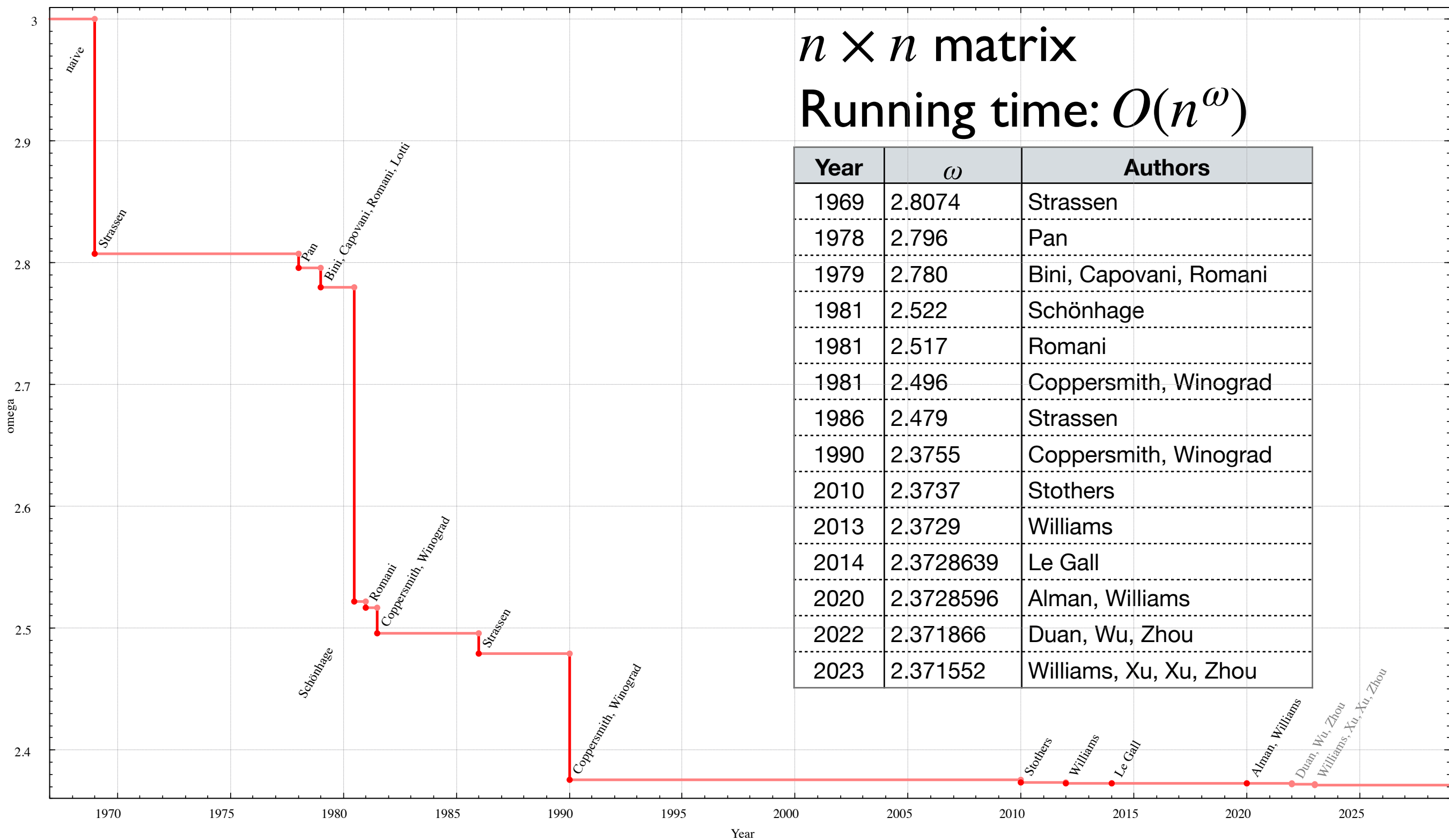
尹一通 Nanjing University, 2023 Fall

Checking Matrix Multiplication

- three $n \times n$ matrices A, B, C :

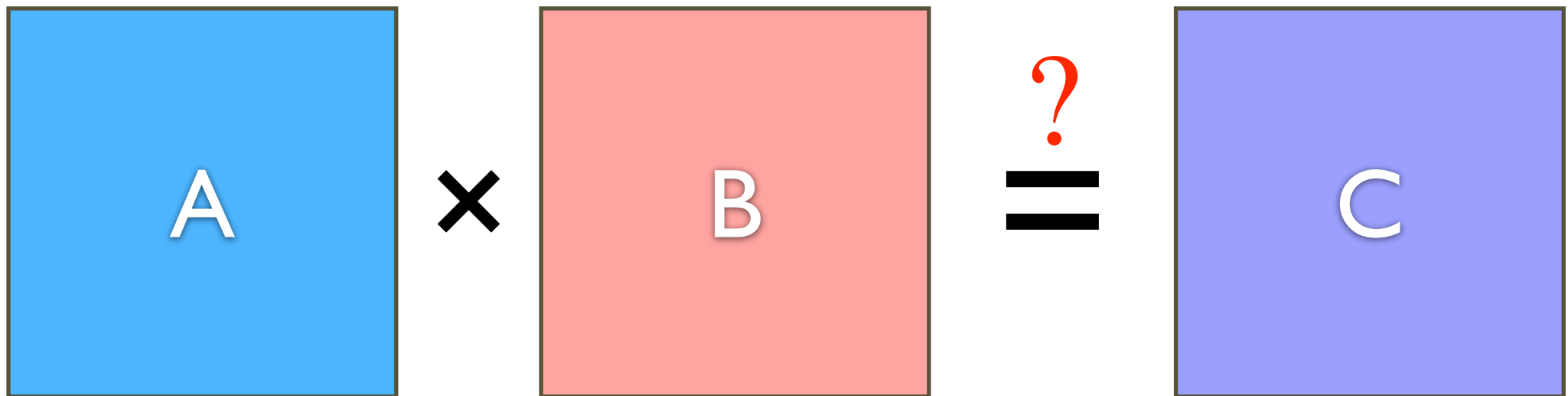


Matrix Multiplication Algorithms



Checking Matrix Multiplication

- three $n \times n$ matrices A, B, C :



Freivald's Algorithm:

pick a uniform random $r \in \{0,1\}^n$;
check whether $A(Br) = Cr$;

time: $O(n^2)$

if $AB = C$: always correct

if $AB \neq C$:

Freivald's Algorithm:

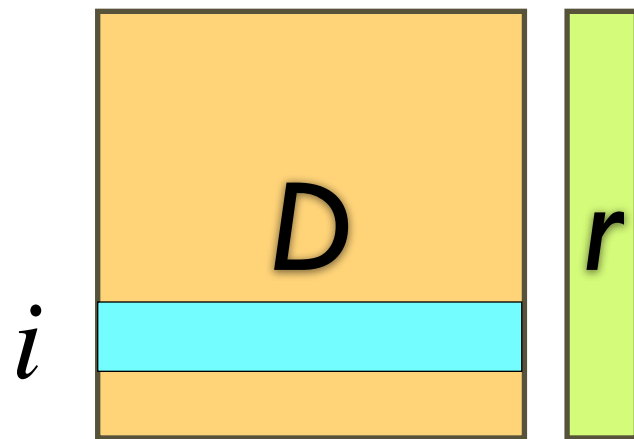
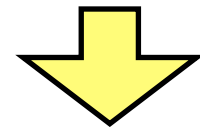
pick a uniform random $r \in \{0,1\}^n$;

check whether $A(Br) = Cr$;

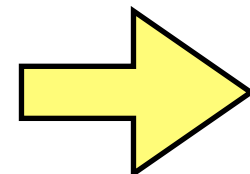
if $AB \neq C$: let $D = AB - C \neq \mathbf{0}_{n \times n}$

suppose $D_{ij} \neq 0$

$$\Pr[ABr = Cr] = \Pr[Dr = \mathbf{0}] \leq \frac{2^{n-1}}{2^n} = \frac{1}{2}$$



$$(Dr)_i = \sum_{k=1}^n D_{ik} r_k = 0$$


$$r_j = -\frac{1}{D_{ij}} \sum_{k \neq j} D_{ik} r_k$$

Freivald's Algorithm:

pick a uniform random $r \in \{0,1\}^n$;
check whether $A(Br) = Cr$;

if $AB = C$: always correct

Theorem (Freivald 1979).

For $n \times n$ matrices A, B, C , if $AB \neq C$, for uniform random $r \in \{0,1\}^n$,

$$\Pr[ABr = Cr] \leq \frac{1}{2}$$

repeat independently for $O(\log n)$ times

Total running time: $O(n^2 \log n)$

Correct *with high probability* (w.h.p.).

Polynomial Identity Testing (PIT)

Input: two polynomials $f, g \in \mathbb{F}[x]$ of degree d .

Output: $f \equiv g$?

$\mathbb{F}[x]$: **polynomial ring** in x over **field** $\mathbb{F}$

$f \in \mathbb{F}[x]$ of degree d : $f(x) = \sum_{i=0}^d a_i x^i$ where $a_i \in \mathbb{F}$ 

Input: a polynomial $f \in \mathbb{F}[x]$ of degree d .

Output: $f \equiv 0$?

f is given as **black-box**

Input: a polynomial $f \in \mathbb{F}[x]$ of degree d .

Output: $f \equiv 0$?

- Deterministic algorithm (**polynomial interpolation**):

pick arbitrary *distinct* $x_0, x_1, \dots, x_d \in \mathbb{F}$;
check if $f(x_i) = 0$ for all $0 \leq i \leq d$;

Fundamental Theorem of Algebra.

Any non-zero d -degree polynomial $f \in \mathbb{F}[x]$ has at most d roots.

- Randomized algorithm (**fingerprinting**):

pick a uniform random $r \in S$;
check if $f(r) = 0$;

let $S \subseteq \mathbb{F}$ be arbitrary
(whose size to be fixed later)

Input: a polynomial $f \in \mathbb{F}[x]$ of degree d .

Output: $f \equiv 0$?

pick a uniform random $r \in S$;
check if $f(r) = 0$;

let $S \subseteq \mathbb{F}$ be arbitrary
~~(whose size to be fixed later)~~

$$|S| = 2d$$

if $f \equiv 0$: always correct

if $f \not\equiv 0$:

$$\Pr[f(r) = 0] \leq \frac{d}{|S|} = \frac{1}{2}$$

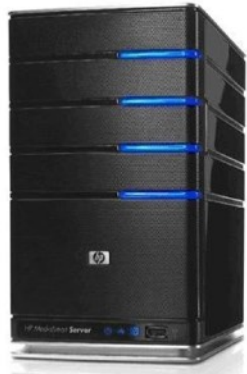
Fundamental Theorem of Algebra.

Any non-zero d -degree polynomial $f \in \mathbb{F}[x]$ has at most d roots.

Checking Identity

北京

database 1



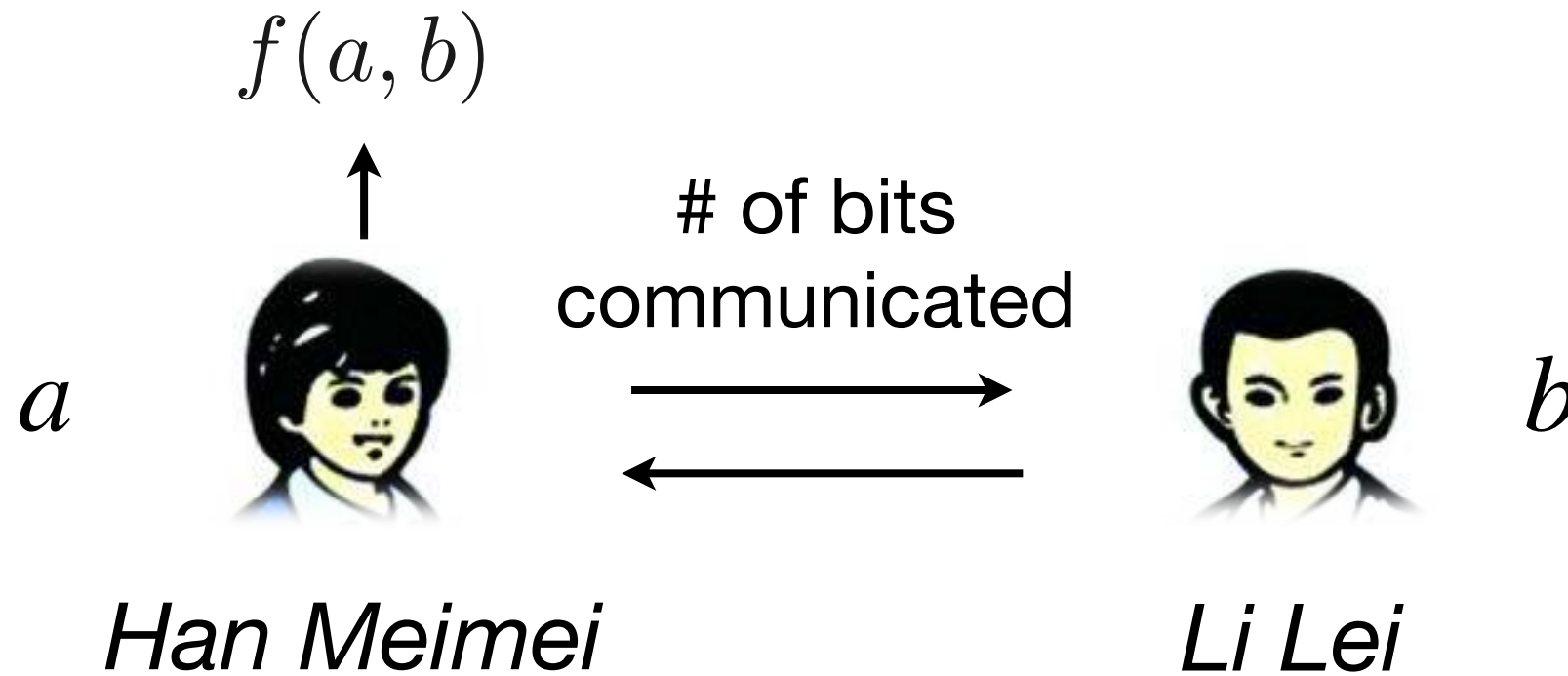
Are they
identical?

南京



database 2

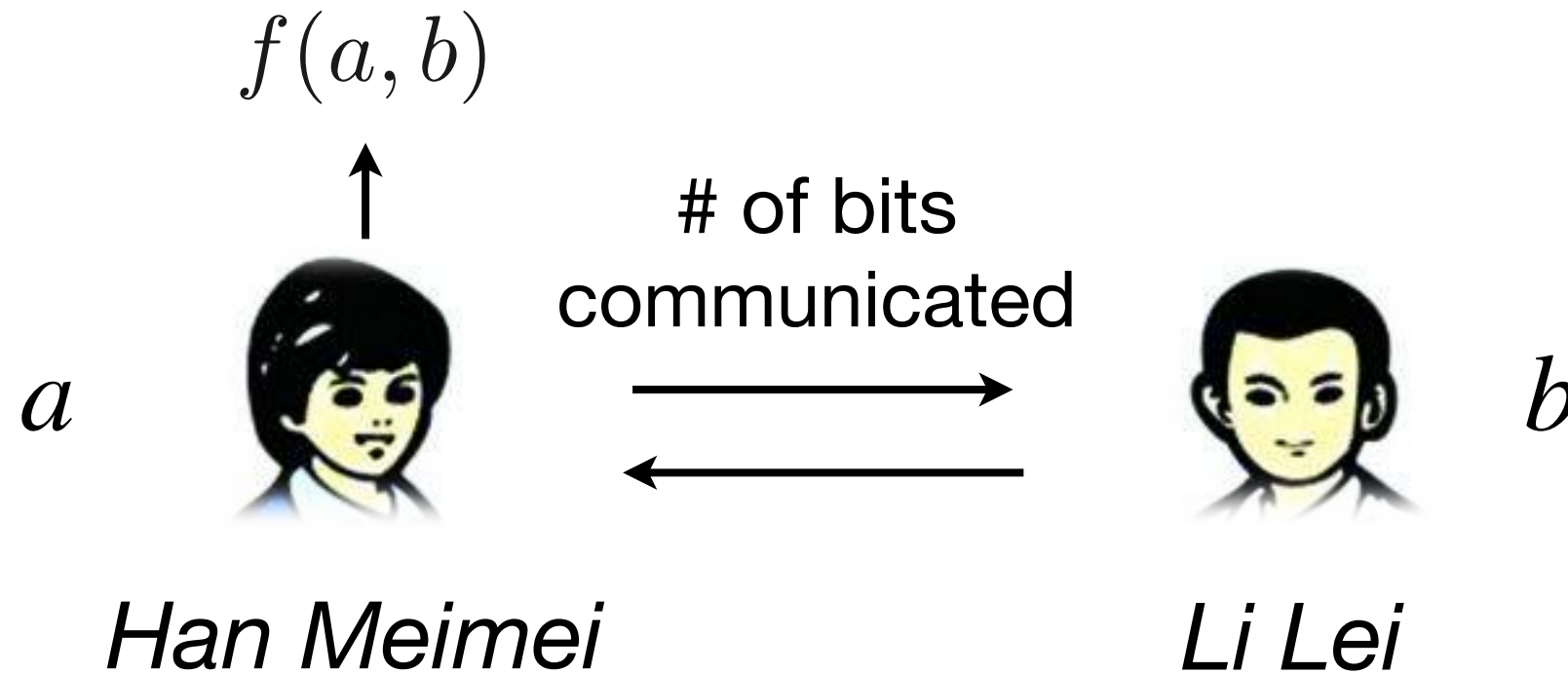
Communication Complexity



$$\text{EQ} : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$$

$$\text{EQ}(a, b) = \begin{cases} 1 & a = b \\ 0 & a \neq b \end{cases}$$

Communication Complexity

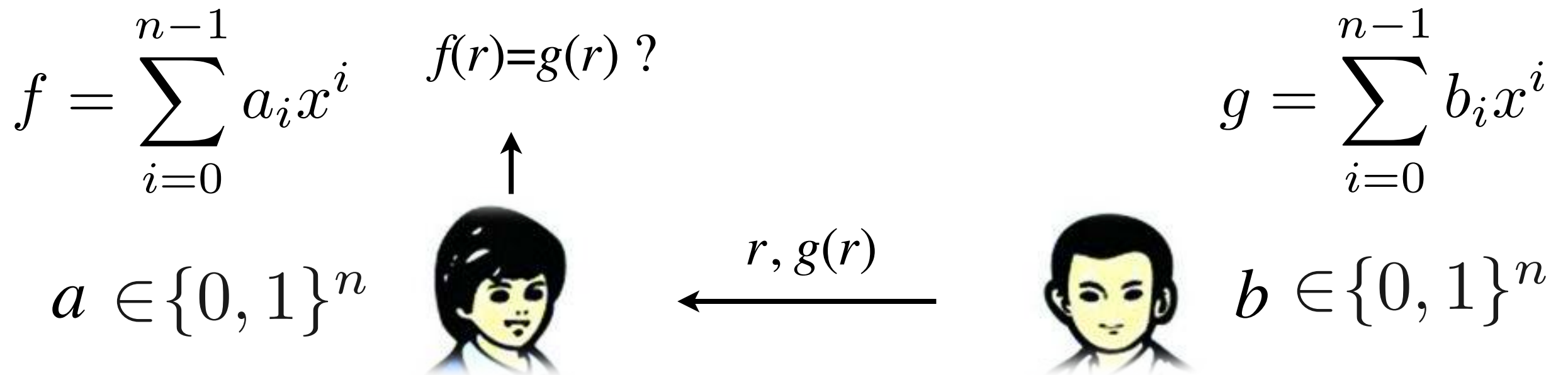


$$\text{EQ} : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$$

Theorem (Yao 1979).

Every deterministic communication protocol solving *EQ* communicates n bits in the worst-case.

Communication Complexity



by **PIT**:

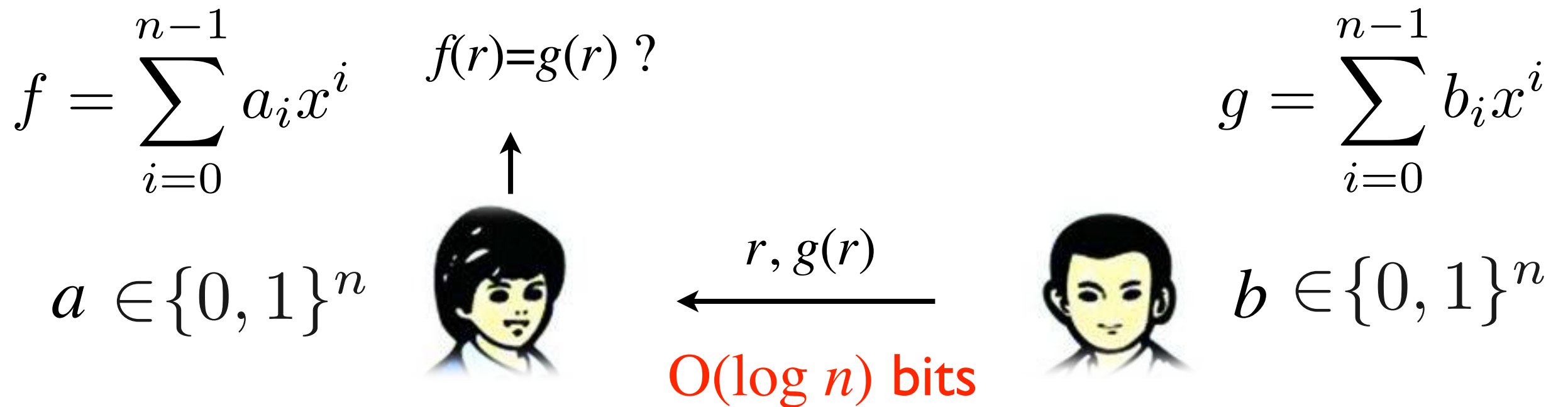
one-sided error $\leq \frac{1}{2}$

pick uniform
random $r \in [2n]$

of bit communicated:

too large!

Communication Complexity



pick uniform
random $r \in [p]$

- choose a prime $p \in [n^2, 2n^2]$
- let $f, g \in \mathbb{Z}_p[x]$
- by PIT: one-sided error is $\frac{n}{p} = O\left(\frac{1}{n}\right)$ (correct w.h.p.)

Polynomial Identity Testing (PIT)

Input: a polynomial $f \in \mathbb{F}[x_1, \dots, x_n]$ of degree d .

Output: $f \equiv 0$?

$\mathbb{F}[x_1, \dots, x_n]$: ring of n -variate polynomials in $x_1, \dots, x_n$ over **field** $\mathbb{F}$

$f \in \mathbb{F}[x_1, \dots, x_n]$:

$$f(x_1, \dots, x_n) = \sum_{i_1, \dots, i_n \geq 0} a_{i_1, i_2, \dots, i_n} x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$$

Degree of f : maximum $i_1 + i_2 + \cdots + i_n$ with $a_{i_1, i_2, \dots, i_n} \neq 0$

Polynomial Identity Testing (PIT)

Input: a polynomial $f \in \mathbb{F}[x_1, \dots, x_n]$ of degree d .

Output: $f \equiv 0$?

$$f(x_1, \dots, x_n) = \sum_{\substack{i_1, \dots, i_n \geq 0 \\ i_1 + \dots + i_n \leq d}} a_{i_1, i_2, \dots, i_n} x_1^{i_1} x_2^{i_2} \dots x_n^{i_n}$$

f is given as **black-box**: given any $\vec{x} \in \mathbb{F}^n$, return $f(\vec{x})$

or as **product form**: e.g. Vandermonde determinant

$$M = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^{n-1} \end{bmatrix} \quad f(\vec{x}) = \det(M) = \prod_{j < i} (x_i - x_j)$$

Polynomial Identity Testing (PIT)

Input: a polynomial $f \in \mathbb{F}[x_1, \dots, x_n]$ of degree d .

Output: $f \equiv 0$?

f is given as **product form**

if $\exists$ a *poly-time deterministic* algorithm for PIT:



either: **NEXP $\neq$ P/poly**

or: **#P $\neq$ FP**

Input: a polynomial $f \in \mathbb{F}[x_1, \dots, x_n]$ of degree d .

Output: $f \equiv 0$?

Fix an arbitrary $S \subseteq \mathbb{F}$:

pick $r_1, \dots, r_n \in S$ uniformly and independently at random;
check if $f(r_1, \dots, r_n) = 0$;

$$f \equiv 0 \implies f(r_1, \dots, r_n) = 0$$

Schwartz-Zippel Theorem.

$$f \not\equiv 0 \implies \Pr [f(r_1, \dots, r_n) = 0] \leq \frac{d}{|S|}$$

of roots for any $f \not\equiv 0$ in any cube S^n is $\leq d \cdot |S|^{n-1}$

Schwartz-Zippel Theorem.

$$f \neq 0 \implies \Pr [f(r_1, \dots, r_n) = 0] \leq \frac{d}{|S|}$$

$$f(x_1, x_2, \dots, x_n) = \sum_{\substack{i_1, i_2, \dots, i_n \geq 0 \\ i_1 + i_2 + \dots + i_n \leq d}} a_{i_1, i_2, \dots, i_n} x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$$

f can be treated as a single-variate polynomial of x_n :

$$\begin{aligned} f(x_1, x_2, \dots, x_n) &= \sum_{i=0}^d x_n^i f_i(x_1, x_2, \dots, x_{n-1}) \\ &= g_{x_1, x_2, \dots, x_{n-1}}(x_n) \end{aligned}$$

$$\Pr[f(r_1, r_2, \dots, r_n) = 0] = \Pr[g_{r_1, r_2, \dots, r_{n-1}}(r_n) = 0]$$

$$g_{r_1, r_2, \dots, r_{n-1}} \neq 0?$$

done?

Schwartz-Zippel Theorem.

$$f \not\equiv 0 \implies \Pr [f(r_1, \dots, r_n) = 0] \leq \frac{d}{|S|}$$

induction on n :

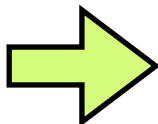
basis: $n=1$ single-variate case, proved by
the *fundamental Theorem of algebra*

I.H.: Schwartz-Zippel Thm is true for all smaller n

Schwartz-Zippel Theorem.

$$f \not\equiv 0 \implies \Pr [f(r_1, \dots, r_n) = 0] \leq \frac{d}{|S|}$$

induction step:

k : highest power of x_n in f  $\begin{cases} f_k \not\equiv 0 \\ \text{degree of } f_k \leq d - k \end{cases}$

$$f(x_1, x_2, \dots, x_n) = \sum_{i=0}^k x_n^i f_i(x_1, x_2, \dots, x_{n-1})$$

$$= x_n^k f_k(x_1, x_2, \dots, x_{n-1}) + \bar{f}(x_1, x_2, \dots, x_n)$$

where $\bar{f}(x_1, x_2, \dots, x_n) = \sum_{i=0}^{k-1} x_n^i f_i(x_1, x_2, \dots, x_{n-1})$

highest power of x_n in $\bar{f} < k$

Schwartz-Zippel Theorem.

$$f \not\equiv 0 \implies \Pr [f(r_1, \dots, r_n) = 0] \leq \frac{d}{|S|}$$

$$f(x_1, x_2, \dots, x_n) = x_n^k f_k(x_1, x_2, \dots, x_{n-1}) + \bar{f}(x_1, x_2, \dots, x_n)$$

$$\begin{cases} f_k \not\equiv 0 \\ \text{degree of } f_k \leq d - k \end{cases}$$

highest power of x_n in $\bar{f} < k$

law of total probability:

$$\Pr[f(r_1, r_2, \dots, r_n) = 0] \quad \text{I.H.} \implies \boxed{} \leq \frac{d - k}{|S|}$$

$$\begin{aligned} &= \Pr[f(\vec{r}) = 0 \mid f_k(r_1, \dots, r_{n-1}) = 0] \cdot \Pr[f_k(r_1, \dots, r_{n-1}) = 0] \\ &+ \Pr[f(\vec{r}) = 0 \mid f_k(r_1, \dots, r_{n-1}) \neq 0] \cdot \Pr[f_k(r_1, \dots, r_{n-1}) \neq 0] \end{aligned}$$

$$\boxed{} = \Pr[g_{r_1, \dots, r_{n-1}}(r_n) = 0 \mid f_k(r_1, \dots, r_{n-1}) \neq 0] \leq \frac{k}{|S|}$$

where $g_{x_1, \dots, x_{n-1}}(x_n) = f(x_1, \dots, x_n)$

Schwartz-Zippel Theorem.

$$f \not\equiv 0 \implies \Pr [f(r_1, \dots, r_n) = 0] \leq \frac{d}{|S|}$$

$$\Pr[f(r_1, r_2, \dots, r_n) = 0] \leq \frac{d - k}{|S|} + \frac{k}{|S|} = \frac{d}{|S|}$$

Input: a polynomial $f \in \mathbb{F}[x_1, \dots, x_n]$ of degree d .

Output: $f \equiv 0$?

Fix an arbitrary $S \subseteq \mathbb{F}$:

pick $r_1, \dots, r_n \in S$ uniformly and independently at random;
check if $f(r_1, \dots, r_n) = 0$;

$$f \equiv 0 \implies f(r_1, \dots, r_n) = 0$$

Schwartz-Zippel Theorem.

$$f \not\equiv 0 \implies \Pr [f(r_1, \dots, r_n) = 0] \leq \frac{d}{|S|}$$

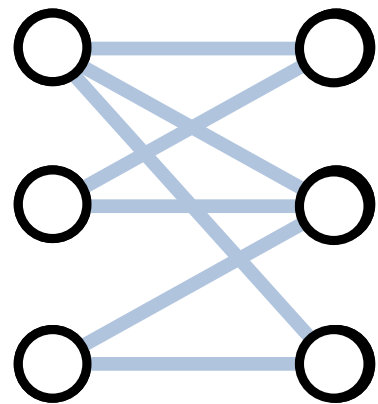
of roots for any $f \not\equiv 0$ in any cube S^n is $\leq d \cdot |S|^{n-1}$

Applications of *Schwartz-Zippel*

- test whether a graph has perfect matching;
- test isomorphism of rooted trees;
- distance property of Reed-Muller codes;
- proof of hardness vs randomness tradeoff;
- algebraic construction of *probabilistically checkable proofs* (PCP);
-

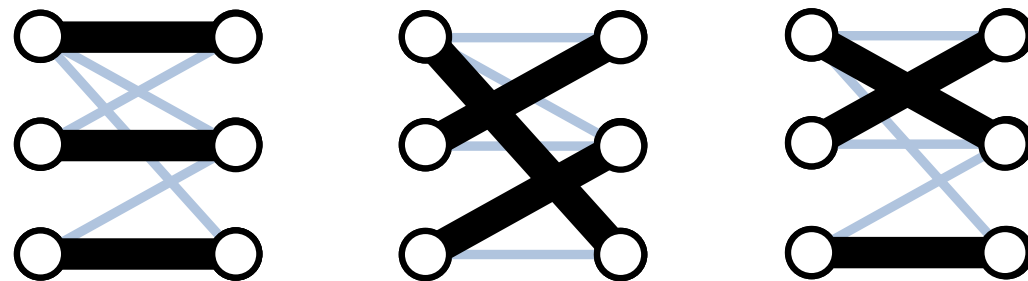
Bipartite Perfect Matching

bipartite graph

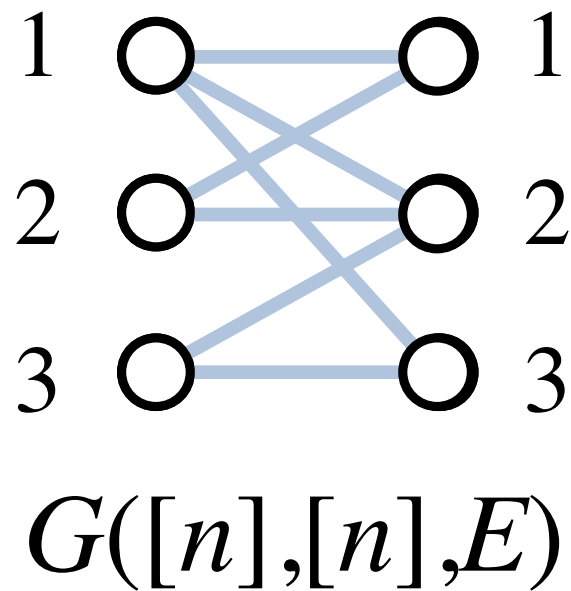


$$G([n],[n],E)$$

perfect matchings



- determine whether G has a perfect matching:
 - Hall's theorem: enumerates all subset of $[n]$
 - Hungarian method: $O(n^3)$
 - Hopcroft-Karp algorithm: $O(m\sqrt{n})$



$$A = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & 0 \\ 0 & x_{32} & x_{33} \end{bmatrix}$$

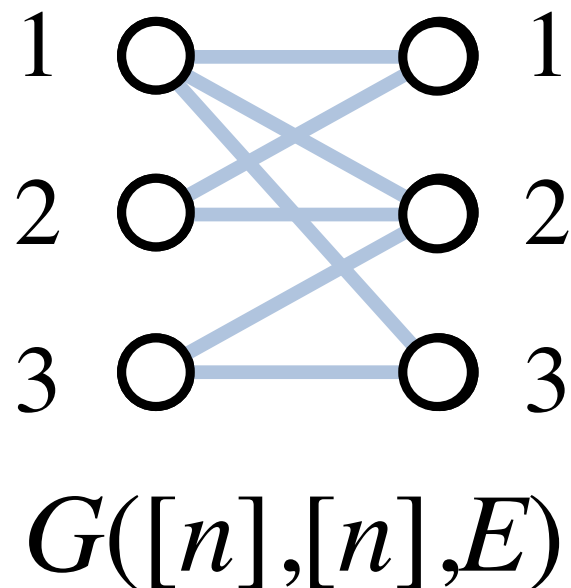
$$\det(A) = x_{11}x_{22}x_{33} + x_{13}x_{21}x_{32} - x_{12}x_{21}x_{33}$$

Edmonds matrix: an $n \times n$ matrix A defined as

$$\forall i, j \in [n], \quad A(i, j) = \begin{cases} x_{i,j} & \text{if } (i, j) \in E \\ 0 & \text{if } (i, j) \notin E \end{cases}$$

Theorem: $\det(A) \neq 0 \iff \exists$ a perfect matching in G

$$\det(A) = \sum_{\pi \in S_n} \text{sgn}(\pi) \prod_{i \in [n]} A(i, \pi(i)) = \sum_{\pi \in S_n} \text{sgn}(\pi) \begin{cases} \prod_{i \in [n]} x_{i, \pi(i)} & \pi \text{ is a P.M.} \\ 0 & \text{otherwise} \end{cases}$$



$$A = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & 0 \\ 0 & x_{32} & x_{33} \end{bmatrix}$$

$$\det(A) = x_{11}x_{22}x_{33} + x_{13}x_{21}x_{32} - x_{12}x_{21}x_{33}$$

Edmonds matrix: an $n \times n$ matrix A defined as

$$\forall i, j \in [n], \quad A(i, j) = \begin{cases} x_{i,j} & \text{if } (i, j) \in E \\ 0 & \text{if } (i, j) \notin E \end{cases}$$

Theorem: $\det(A) \neq 0 \iff \exists$ a perfect matching in G

- $\det(A)$ is an m -variate degree- n polynomial:
 - Use *Schwartz-Zippel* to check whether $\det(A) \neq 0$
 - Computing determinants is generic and can be done in parallel (Chistov's algorithm)

Fingerprinting

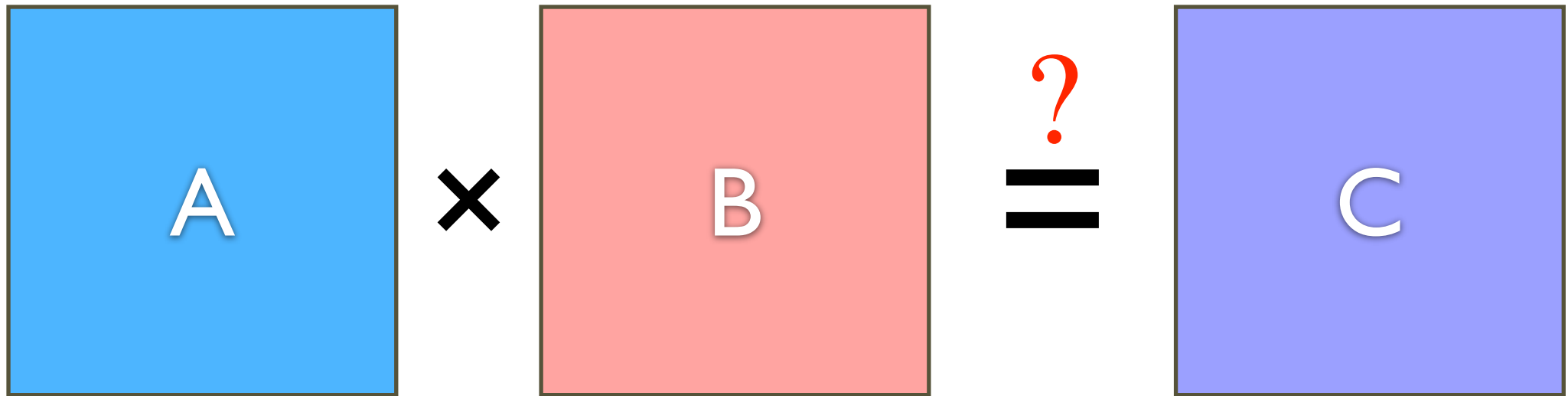


$$\begin{array}{ccc} X & = & Y \quad ? \\ \downarrow & & \downarrow \\ \text{FING}(X) & = & \text{FING}(Y) \quad ? \end{array}$$

- $\text{FING}()$ is a function: $X = Y \implies \text{FING}(X) = \text{FING}(Y)$
- if $X \neq Y$, $\Pr[\text{FING}(X) = \text{FING}(Y)]$ is small.
- Fingerprints are easy to compute and compare.

Checking Matrix Multiplication

- three $n \times n$ matrices A, B, C :



Freivald's Algorithm:

pick a uniform random $r \in \{0,1\}^n$;
check whether $A(Br) = Cr$;

For an $n \times n$ matrix M :

$$\text{FING}(M) = Mr \text{ for uniform random } r \in \{0,1\}^n$$

Polynomial Identity Testing (PIT)

Input: a polynomial $f \in \mathbb{F}[x_1, \dots, x_n]$ of degree d .

Output: $f \equiv 0$?

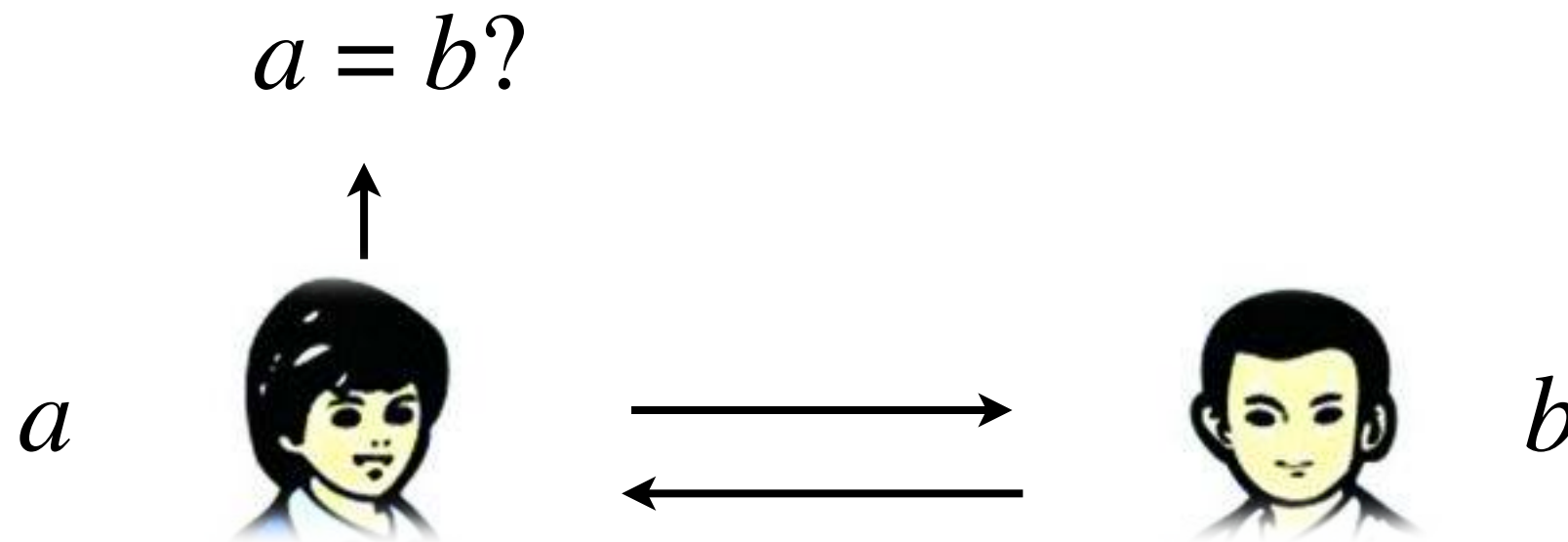
Fix an arbitrary $S \subseteq \mathbb{F}$:

pick $r_1, \dots, r_n \in S$ uniformly and independently at random;
check if $f(r_1, \dots, r_n) = 0$;

For a polynomial $f \in \mathbb{F}[x_1, \dots, x_n]$:

FING(f) = $f(r_1, \dots, r_n)$ for uniform independent $r_1, \dots, r_n \in S$

Communication Complexity

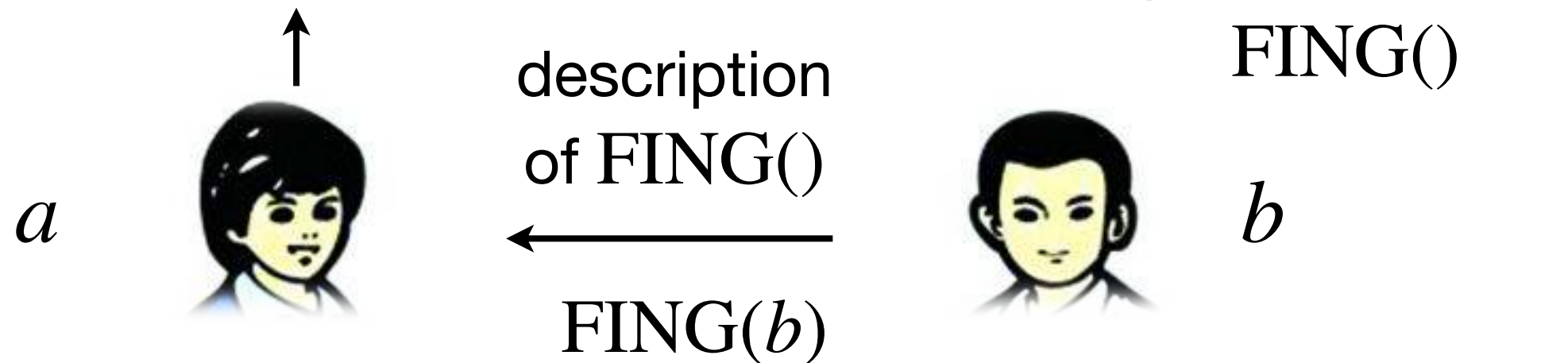


$$\text{EQ} : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}$$

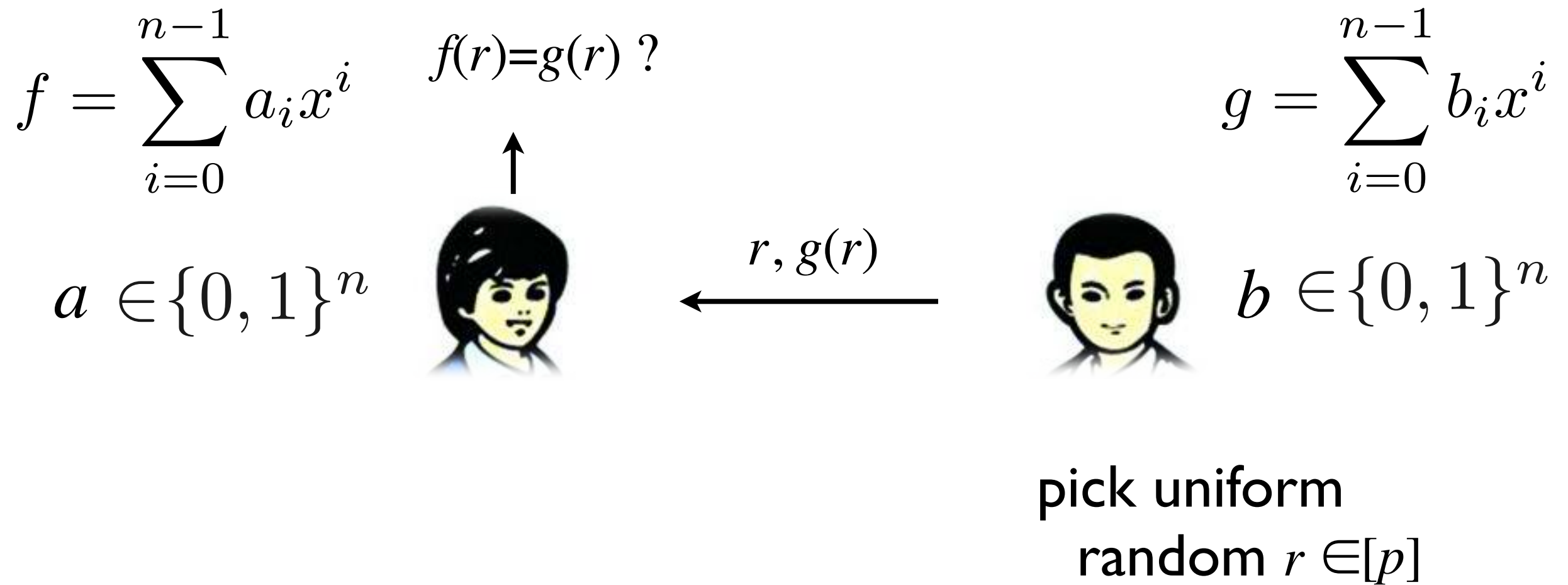
$$\text{EQ}(a, b) = \begin{cases} 1 & a = b \\ 0 & a \neq b \end{cases}$$

Fingerprinting

$$\text{FING}(a) = \text{FING}(b)?$$

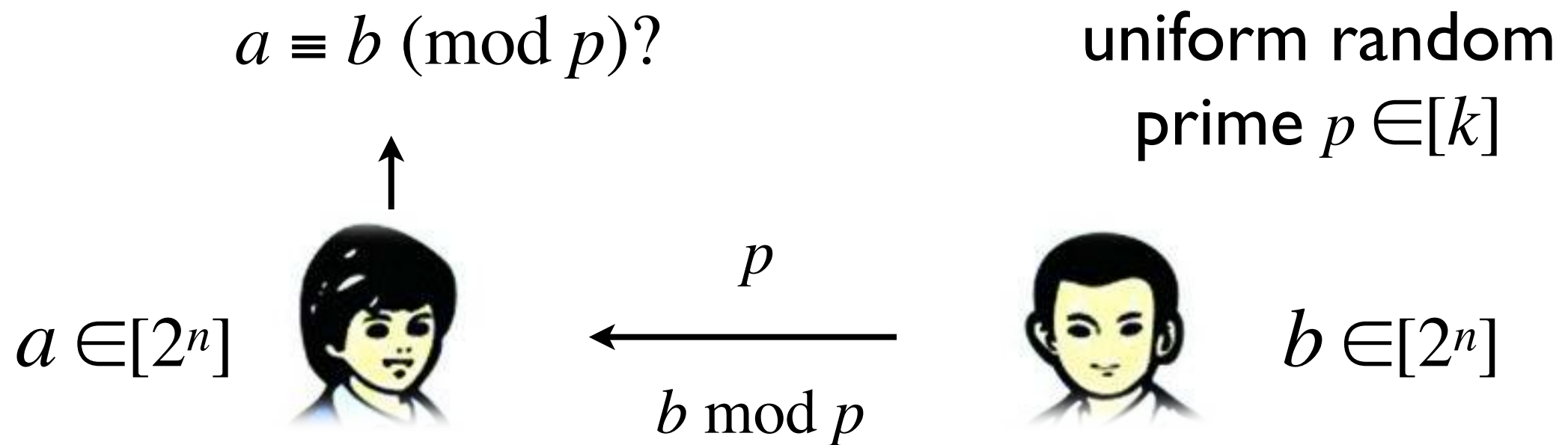


- $\text{FING}()$ is a function: $a = b \implies \text{FING}(a) = \text{FING}(b)$
- if $a \neq b$, $\Pr[\text{FING}(a) = \text{FING}(b)]$ is small.
- Fingerprints are short.



$f, g \in \mathbb{Z}_p[x]$ for a prime $p \in [n^2, 2n^2]$

$$\text{FING}(b) = \sum_{i=0}^{n-1} b_i r^i \text{ for random } r$$



$\text{FING}(x) = x \bmod p$ for uniform random prime $p \in [k]$

communication complexity: $O(\log k)$

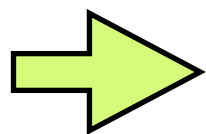
if $a = b \Rightarrow a \equiv b \pmod{p}$

if $a \neq b$: $\Pr[a \equiv b \pmod{p}] \leq ?$

for a $z = |a - b| \neq 0$: $\Pr[z \bmod p = 0] \leq ?$

uniform random prime $p \in [k]$

for a $z = |a - b| \neq 0$: $\Pr[z \bmod p = 0] \leq ?$

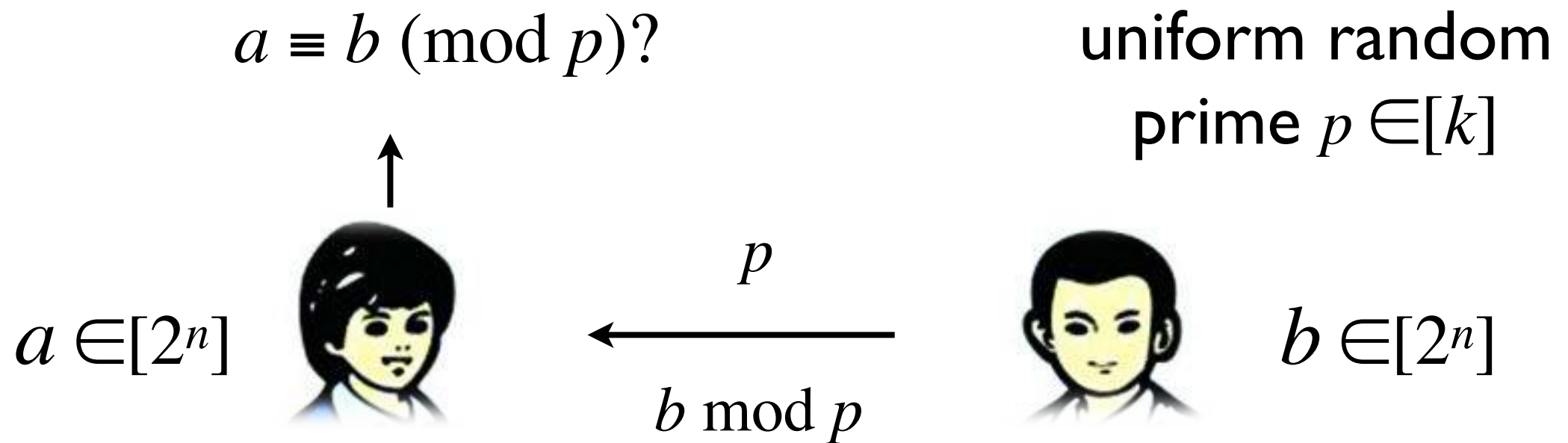
$\in [2^n]$
each prime divisor ≥ 2 }  # of prime divisors of $z \leq n$

$$\Pr[z \bmod p = 0] = \frac{\text{\# of prime divisors of } z \leq n}{\text{\# of primes in } [k]} = \pi(k)$$

$\pi(N)$: # of primes in $[N]$

Prime Number Theorem (PNT):

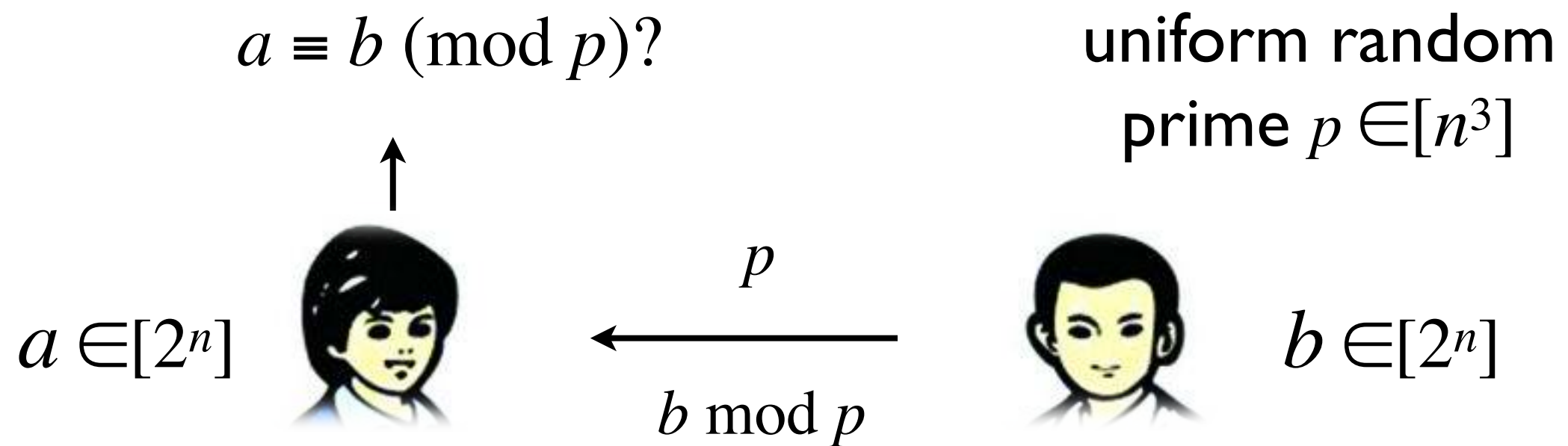
$$\pi(N) \sim \frac{N}{\ln N} \text{ as } N \rightarrow \infty$$



for a $z = |a - b| \neq 0$: $\Pr[z \bmod p = 0] \leq ?$

$$\Pr[z \bmod p = 0] = \frac{\text{\# of prime divisors of } z \leq n}{\text{\# of primes in } [k] = \pi(k)}$$

choose $k = n^3$ $\leq \frac{n \ln k}{k} = \frac{3 \ln n}{n^2} = O\left(\frac{1}{n}\right)$



$\text{FING}(b) = b \bmod p$ for uniform random prime $p \in [n^3]$

communication complexity: $O(\log n)$

if $a = b \Rightarrow a \equiv b \pmod{p}$

if $a \neq b \Rightarrow \Pr[a \equiv b \pmod{p}] = O\left(\frac{1}{n}\right)$

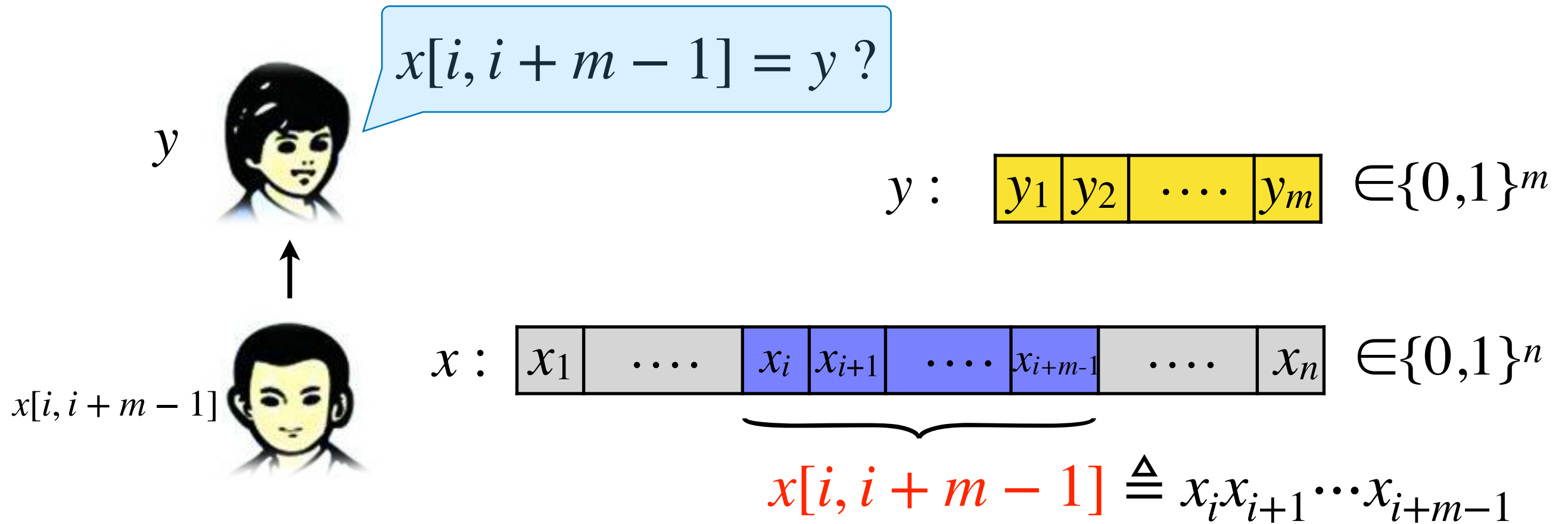
Pattern Matching

Input: string $x \in \{0,1\}^n$, **pattern** $y \in \{0,1\}^m$

Check whether y is a substring of x .

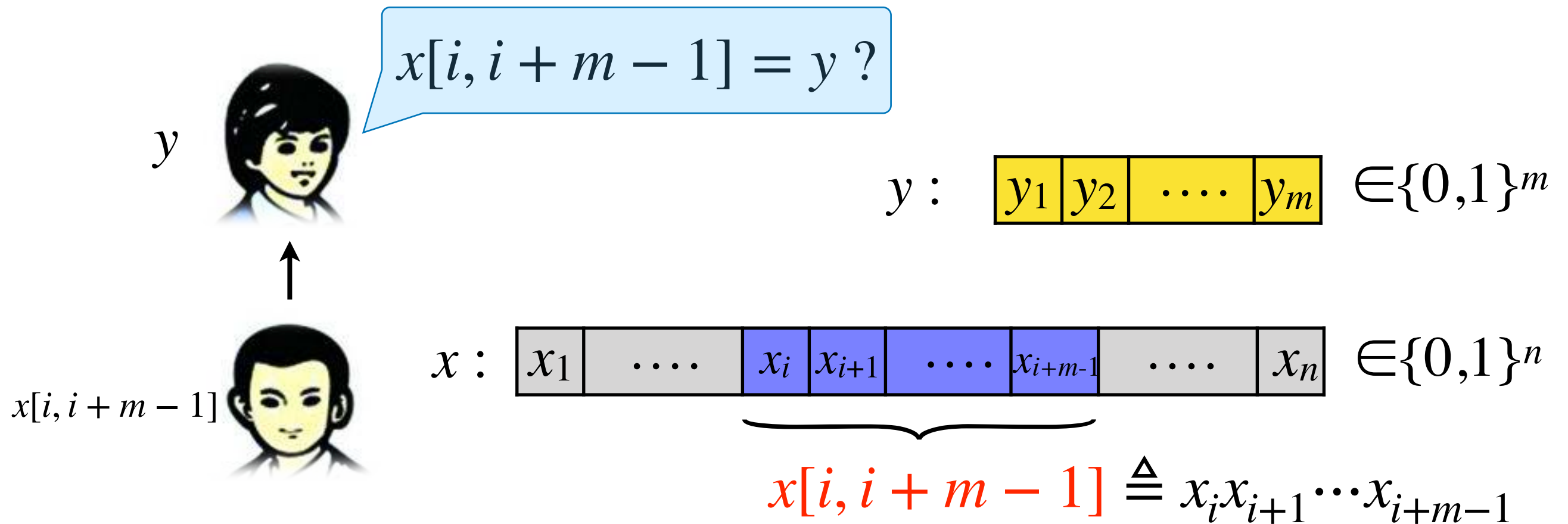
- naive algorithm: $O(mn)$ time
- Knuth-Morris-Prat (**KMP**) algorithm: $O(m + n)$ time
 - finite state automaton

Pattern Matching via Fingerprinting



```
pick a random FING();  
for  $i = 1, 2, \dots, n - m + 1$  do:  
    if  $\text{FING}(x[i, i + m - 1]) = \text{FING}(y)$  then return  $i$ ;  
return "no match";
```

Karp-Rabin Algorithm



Karp-Rabin Algorithm: $\text{FING}(a) = a \bmod p$

pick a uniform random prime $p \in [mn^3]$;

for $i = 1, 2, \dots, n - m + 1$ do:

 if $x[i, i + m - 1] \equiv y \pmod{p}$ then return i ;

return “no match”;

$$y : \begin{array}{|c|c|c|c|} \hline y_1 & y_2 & \cdots & y_m \\ \hline \end{array} \in \{0,1\}^m$$

$$x : \begin{array}{|c|c|c|c|c|c|c|c|} \hline x_1 & \cdots & x_i & x_{i+1} & \cdots & x_{i+m-1} & \cdots & x_n \\ \hline \end{array} \in \{0,1\}^n$$

Karp-Rabin Algorithm: $\text{FING}(a) = a \bmod p$

pick a uniform random prime $p \in [mn^3]$;

for $i = 1, 2, \dots, n - m + 1$ do:

 if $x[i, i + m - 1] \equiv y \pmod{p}$ then return i ;

return “no match”;

For each i , if $x[i, i + m - 1] \neq y$:

$$\Pr [x[i, i + m - 1] \equiv y \pmod{p}] \leq m \ln(mn^3)/mn^3 = o(1/n^2)$$

By union bound: when y is not a substring of x

$$\begin{aligned} & \Pr[\text{the algorithm ever makes a mistake}] \\ & \leq \Pr [\exists i, x[i, i + m - 1] \equiv y \pmod{p}] = o(1/n) \end{aligned}$$

$$y : \begin{array}{|c|c|c|c|} \hline y_1 & y_2 & \cdots & y_m \\ \hline \end{array} \in \{0,1\}^m$$

$$x : \begin{array}{|c|c|c|c|c|c|c|c|} \hline x_1 & \cdots & x_i & x_{i+1} & \cdots & x_{i+m-1} & \cdots & x_n \\ \hline \end{array} \in \{0,1\}^n$$

$$\underbrace{x_i x_{i+1} \cdots x_{i+m-1}}_{x[i, i+m-1]} \triangleq x_i x_{i+1} \cdots x_{i+m-1}$$

Karp-Rabin Algorithm: $\text{FING}(a) = a \bmod p$
 pick a uniform random prime $p \in [mn^3]$;
 for $i = 1, 2, \dots, n - m + 1$ do:
 if $x[i, i+m-1] \equiv y \pmod{p}$ then return i ;
 return “no match”;
Testable in $O(1)$ time

Observe: $x[i+1, i+m] = x_{i+m} + 2 \left(x[i, i+m-1] - 2^{m-1} x_i \right)$

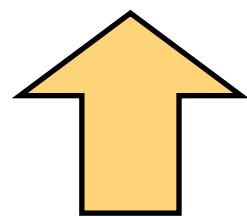


$$\text{FING}(x[i+1, i+m]) = \left(x_{i+m} + 2 \left(\text{FING}(x[i, i+m-1]) - 2^{m-1} x_i \right) \right) \bmod p$$

Checking Distinctness

Input: n numbers $x_1, x_2, \dots, x_n \in \{1, 2, \dots, n\}$

Determine whether every number appears **exactly once**.

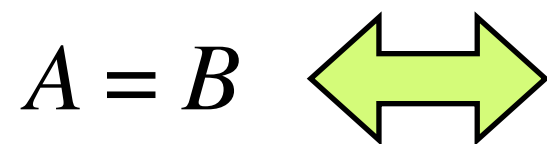


$$A = \{x_1, x_2, \dots, x_n\}$$

$$B = \{1, 2, \dots, n\}$$

Input: two multisets $A = \{a_1, \dots, a_n\}$ and $B = \{b_1, \dots, b_n\}$
where $a_1, \dots, a_n, b_1, \dots, b_n \in \{1, \dots, n\}$

Output: $A = B$ (as multisets)?

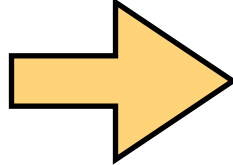


$$A = B \iff \forall x: \begin{array}{l} \# \text{ of times } x \text{ appearing in } A \\ = \# \text{ of times } x \text{ appearing in } B \end{array}$$

Input: two multisets $A = \{a_1, \dots, a_n\}$ and $B = \{b_1, \dots, b_n\}$
where $a_1, \dots, a_n, b_1, \dots, b_n \in \{1, \dots, n\}$

Output: $A = B$ (as multisets)?

- naive algorithm: use $O(n)$ time and $O(n)$ space
- **fingerprinting**: random fingerprint function $\text{FING}()$
 - check $\text{FING}(A) = \text{FING}(B)$?
 - time cost: time to compute and check fingerprints $O(n)$
 - **space cost: space to store fingerprints $O(\log p)$**

multisets $A = \{a_1, a_2, \dots, a_n\}$  $f_A(x) = \prod_{i=1}^n (x - a_i)$

$f_A \in \mathbb{Z}_p[x]$ for prime p (to be specified)

$$\text{FING}(A) = f_A(r) \quad \text{for uniform random } r \in \mathbb{Z}_p$$

multisets $A = \{a_1, a_2, \dots, a_n\}$
 $B = \{b_1, b_2, \dots, b_n\}$
 where $a_i, b_i \in \{1, 2, \dots, n\}$

$\Rightarrow \begin{cases} f_A(x) = \prod_{i=1}^n (x - a_i) \\ f_B(x) = \prod_{i=1}^n (x - b_i) \end{cases}$

$f_A, f_B \in \mathbb{Z}_p[x]$ for prime p (to be specified)

$\left. \begin{array}{l} \text{FING}(A) = f_A(r) \\ \text{FING}(B) = f_B(r) \end{array} \right\}$ for uniform random $r \in \mathbb{Z}_p$

$A \neq B \implies f_A \not\equiv f_B \text{ on reals } \mathbb{R}$

(but possibly $f_A \equiv f_B$ on finite field $\mathbb{Z}_p$)

if $A = B$: $\text{FING}(A) = \text{FING}(B)$

if $A \neq B$: $\text{FING}(A) \neq \text{FING}(B)$

$\Rightarrow \left\{ \begin{array}{l} \bullet f_A \equiv f_B \text{ on finite field } \mathbb{Z}_p \Rightarrow \begin{array}{l} \text{in } f_A - f_B \text{ on } \mathbb{R}: \\ \exists \text{ coefficient } c \neq 0 \\ c \bmod p = 0 \end{array} \\ \bullet f_A \not\equiv f_B \text{ on } \mathbb{Z}_p \text{ but } f_A(r) = f_B(r) \xrightarrow{\text{Schwartz-Zippel}} \text{with probability } \leq n/p \end{array} \right.$

multisets $A = \{a_1, a_2, \dots, a_n\}$
 $B = \{b_1, b_2, \dots, b_n\}$
 where $a_i, b_i \in \{1, 2, \dots, n\}$

$\Rightarrow \begin{cases} f_A(x) = \prod_{i=1}^n (x - a_i) \\ f_B(x) = \prod_{i=1}^n (x - b_i) \end{cases}$

$f_A, f_B \in \mathbb{Z}_p[x]$ for uniform random prime $p \in [L, U]$
 (L, U to be specified)

$\left. \begin{array}{l} \text{FING}(A) = f_A(r) \\ \text{FING}(B) = f_B(r) \end{array} \right\}$ for uniform random $r \in \mathbb{Z}_p$

if $A \neq B$: $\text{FING}(A) = \text{FING}(B)$

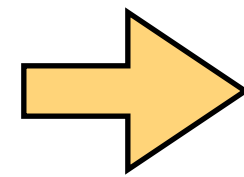
in $f_A - f_B$ on $\mathbb{R}$:
 $\exists$ coefficient $c \neq 0$
 $c \bmod p = 0$

$\Rightarrow \left\{ \begin{array}{l} \bullet f_A \equiv f_B \text{ on finite field } \mathbb{Z}_p \Rightarrow \\ \Pr[c \bmod p = 0] \leq \frac{\# \text{ of prime factors of } c}{\# \text{ of primes in } [L, U]} \\ \quad \boxed{|c| \leq n^n} \Rightarrow \leq \frac{n \log_2 n}{\pi(U) - \pi(L)} \sim \frac{n \log_2 n}{U / \ln U - L / \ln L} \\ \bullet f_A \not\equiv f_B \text{ on } \mathbb{Z}_p \text{ but } f_A(r) = f_B(r) \xrightarrow{\text{Schwartz-Zippel}} \text{with probability } \leq n/p \leq n/L \end{array} \right.$

multisets $A = \{a_1, a_2, \dots, a_n\}$

$B = \{b_1, b_2, \dots, b_n\}$

where $a_i, b_i \in \{1, 2, \dots, n\}$



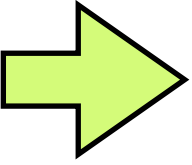
$$\begin{cases} f_A(x) = \prod_{i=1}^n (x - a_i) \\ f_B(x) = \prod_{i=1}^n (x - b_i) \end{cases}$$

$f_A, f_B \in \mathbb{Z}_p[x]$ for uniform random prime $p \in [L, U]$

with $U = 2L = (n \log n)^2$

$\left. \begin{array}{l} \text{FING}(A) = f_A(r) \\ \text{FING}(B) = f_B(r) \end{array} \right\}$ for uniform random $r \in \mathbb{Z}_p$

if $A \neq B$: $\text{FING}(A) = \text{FING}(B)$

 $\left\{ \begin{array}{l} \bullet f_A \equiv f_B \text{ on finite field } \mathbb{Z}_p \xrightarrow{\text{with probability}} \leq \frac{n \log_2 n}{U / \ln U - L / \ln L} = O(1/n) \\ \bullet f_A \not\equiv f_B \text{ on } \mathbb{Z}_p \text{ but } f_A(r) = f_B(r) \xrightarrow[\text{Schwartz-Zippel}]{\text{with probability}} \leq n/p \leq n/L = O(1/n) \end{array} \right.$

Input: two multisets $A = \{a_1, \dots, a_n\}$ and $B = \{b_1, \dots, b_n\}$
where $a_1, \dots, a_n, b_1, \dots, b_n \in \{1, \dots, n\}$

Output: $A = B$ (as multisets)?

Lipton's Algorithm (1989):

$$\left. \begin{aligned} \text{FING}(A) &= \prod_{i=1}^n (r - a_i) \bmod p \\ \text{FING}(B) &= \prod_{i=1}^n (r - b_i) \bmod p \end{aligned} \right\} \begin{aligned} &\text{for uniform random prime} \\ &p \in [(n \log n)^2/2, (n \log n)^2] \\ &\text{and uniform random } r \in \mathbb{Z}_p \end{aligned}$$

if $A \neq B$ as multisets:

$$f_A(x) = \prod_{i=1}^n (x - a_i) \bmod p \qquad f_B(x) = \prod_{i=1}^n (x - b_i) \bmod p$$

$$\begin{aligned} &\Pr[\text{FING}(A) = \text{FING}(B)] \\ &\leq \Pr[f_A \equiv f_B] + \Pr[f_A(r) = f_B(r) \mid f_A \not\equiv f_B] = O(1/n) \end{aligned}$$

Input: n numbers $x_1, x_2, \dots, x_n \in \{1, 2, \dots, n\}$

Determine whether every number appears **exactly once**.

Lipton's Algorithm (1989):

$$\left. \begin{array}{l} \text{FING}(A) = \prod_{i=1}^n (r - a_i) \bmod p \\ \text{check if:} \\ \text{FING}(A) = \prod_{i=1}^n (r - i) \bmod p? \end{array} \right\} \begin{array}{l} \text{for uniform random prime} \\ p \in [(n \log n)^2/2, (n \log n)^2] \\ \text{and uniform random } r \in \mathbb{Z}_p \end{array}$$

- time cost: $O(n)$
- space cost: $O(\log n)$
- error probability (**false positive**): $O(1/n)$
- **data stream**: input comes one at a time