Randomized Algorithms
Course Info

• 尹一通
  yitong.yin@gmail.com
  yinyt@nju.edu.cn

• office hour:
  804, Thursday 2-4pm

• course homepage: http://tcs.nju.edu.cn/wiki
Textbooks


References

CLRS

Feller
An Introduction to Probability Theory and Its Applications

Aldous and Fill
Reversible Markov Chains and Random Walks on Graphs

Alon and Spencer
The Probabilistic Method
Randomized Algorithms

“algorithms which use randomness in computation”

Why?

• Simpler.
• Faster.
• Can do impossibles.
• Can give us clever deterministic algorithms.
• Random input.
• Deterministic problem with random nature.
• ... ...
Checking Matrix Multiplication

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

$$\begin{array}{c}
A \\
\times \\
B \\
\end{array} \quad ? \quad \begin{array}{c}
C \\
\end{array}$$

best matrix multiplication algorithm: $O(n^{2.373})$
Checking Matrix Multiplication

three \( n \times n \) matrices \( A, B, C \):

\[
A \times B = \_ \_ \_ = C
\]

Algorithm: \( O(n^{2.373}) \)
Checking Matrix Multiplication

three \( n \times n \) matrices \( A, B, C \):

\[
\begin{array}{ccc}
A & \times & B \\
\end{array}
\]

\( \overset{?}{=} \)

\( C \)

best matrix multiplication algorithm: \( O(n^{2.373}) \)

Freivald’s Algorithm

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

\( \text{time: } O(n^2) \quad \text{if } AB = C, \text{ always correct} \)
Freivald’s Algorithm

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

if \( AB = C \), always correct

if \( AB \neq C \), let \( D = AB - C \neq 0_{n \times n} \)
say \( D_{ij} \neq 0 \)

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

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

\[
r_j = -\frac{1}{D_{ij}} \sum_{k \neq j}^{n} D_{ik}r_k
\]
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)

If \( AB \neq C \), for a uniformly random \( r \in \{0,1\}^n \),

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

repeat independently for 100 times

time: \( O(n^2) \) \quad if \( AB \neq C \), error probability \( \leq 2^{-100} \)
Monte Carlo vs Las Vegas

Two types of randomized algorithms:

**Monte Carlo**
- running time is fixed
- correctness is random

**Las Vegas**
- always correct
- running time is random
Polynomial Identity Testing (PIT)

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

**Output:** $f \equiv g$?

$f \in \mathbb{F}[x]$ of degree $d$:

$$f(x) = \sum_{i=0}^{d} a_i x^i \quad \text{for} \quad a_i \in \mathbb{F}$$

**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:** $f \equiv 0$?

simple deterministic algorithm:

check whether $f(x) = 0$ for all $x \in \{1, 2, \ldots, d + 1\}$

---

**Fundamental Theorem of Algebra:**
A degree $d$ polynomial has at most $d$ roots.

---

**Randomized Algorithm**

| pick a **uniform** random $r \in S$; |
| check whether $f(r) = 0$; |
| $S \subseteq \mathbb{F}$ |
A degree $d$ polynomial has at most $d$ roots.

**Randomized Algorithm**

<table>
<thead>
<tr>
<th>pick a <strong>uniform</strong> random $r \in S$;</th>
</tr>
</thead>
<tbody>
<tr>
<td>check whether $f(r) = 0$ ;</td>
</tr>
</tbody>
</table>

If $f \neq 0$

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

$S \subseteq \mathbb{F}$

$|S| = 2d$

**Fundamental Theorem of Algebra:**

A degree $d$ polynomial has at most $d$ roots.
Checking Identity

Are they identical?

database 1

database 2
Communication Complexity
(Yao 1979)

\[ f(a, b) \]

\[ \uparrow \]

\[ \text{\# of bits communicated} \]

\[ a \rightarrow \] \text{Han Meimei}

\[ b \leftarrow \]

\[ \text{Li Lei} \]

\[ EQ : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\} \]

**Theorem** (Yao, 1979)

There is no deterministic communication protocol solving EQ with less than \( n \) bits in the worst-case.
Communication Complexity

\[ f = \sum_{i=0}^{n-1} a_i x^i \]  \quad f(r) = g(r) ?

\[ a \in \{0, 1\}^n \]

\[ g = \sum_{i=0}^{n-1} b_i x^i \]

\[ b \in \{0, 1\}^n \]

by PIT:
one-sided error \( \leq \frac{1}{2} \)

# of bit communicated: too large!

pick uniform random \( r \in [2n] \)
Communication Complexity

\[ f = \sum_{i=0}^{n-1} a_i x^i \quad f(r) = g(r) \quad ? \]

\[ a \in \{0, 1\}^n \]

\[ b \in \{0, 1\}^n \]

\[ g = \sum_{i=0}^{n-1} b_i x^i \]

k = \lfloor \log_2(2n) \rfloor

choose a prime \( p \in [2^k, 2^{k+1}] \)

let \( f, g \in \mathbb{Z}_p[x] \)

pick uniform random \( r \in [p] \)

\( O(\log n) \) bits

Han Meimei

Li Lei
Polynomial Identity Testing (PIT)

**Input:** \( f, g \in \mathbb{F}[x_1, x_2, \ldots, x_n] \) of degree \( d \)

**Output:** \( f \equiv g? \)

\( \mathbb{F}[x_1, x_2, \ldots, x_n] \) : ring of \( n \)-variate polynomials over field \( \mathbb{F} \)

\( f \in \mathbb{F}[x_1, x_2, \ldots, x_n] \) :

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

degree of \( f \) : maximum \( i_1 + i_2 + \cdots + i_n \) with \( a_{i_1, i_2, \ldots, i_n} \neq 0 \)
Input: \( f, g \in \mathbb{F}[x_1, x_2, \ldots, x_n] \) of degree \( d \)
Output: \( f \equiv g? \)

equivalently:

Input: \( f \in \mathbb{F}[x_1, x_2, \ldots, x_n] \) of degree \( d \)
Output: \( f \equiv 0? \)

\[
f(x_1, x_2, \ldots, x_n) = \sum_{i_1, i_2, \ldots, i_n \geq 0 \atop i_1 + i_2 + \cdots + i_n \leq d} a_{i_1, i_2, \ldots, i_n} x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}
\]
Input: \( f, g \in \mathbb{F}[x_1, x_2, \ldots, x_n] \) of degree \( d \)

Output: \( f \equiv g? \)

equivalently:

Input: \( f \in \mathbb{F}[x_1, x_2, \ldots, x_n] \) of degree \( d \)

Output: \( f \equiv 0? \)

\( f \) is given as block-box: given any \( \vec{x} = (x_1, x_2, \ldots, x_n) \)
returns \( f(\vec{x}) \)

or as product from: e.g. Vandermonde determinant

\[
M = \begin{bmatrix}
1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\
1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & x_n & x_n^2 & \cdots & x_n^{n-1}
\end{bmatrix}
\]

\[
f(\vec{x}) = \det(M) = \prod_{j<i} (x_i - x_j)
\]
PIT: **Polynomial Identity Testing**

**Input:** \( f \in \mathbb{F}[x_1, x_2, \ldots, x_n] \) of degree \( d \)

**Output:** \( f \equiv 0? \)

\( f \) is given as **block-box** or **product from**

If \( \exists \) a **poly-time deterministic** algorithm for PIT:

- either: \( \text{NEXP} \neq \text{P/poly} \)
- or: \( \#P \neq \text{FP} \)
Input: \( f \in \mathbb{F}[x_1, x_2, \ldots, x_n] \) of degree \( d \)

Output: \( f \equiv 0? \)

fix an arbitrary \( S \subseteq \mathbb{F} \)

pick random \( r_1, r_2, \ldots, r_n \in S \)

uniformly and independently at random;

check whether \( f(r_1, r_2, \ldots, r_n) = 0 \);

\[ f \equiv 0 \quad \Rightarrow \quad f(r_1, r_2, \ldots, r_n) = 0 \]
Input: a polynomial \( f \in \mathbb{F}[x] \) of degree \( d \)

Output: \( f \equiv 0 \) ?

fix an arbitrary \( S \subseteq \mathbb{F} \)

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

\[
f \equiv 0 \quad \Rightarrow \quad f(r) = 0
\]

Fundamental Theorem of Algebra:
A degree \( d \) polynomial has at most \( d \) roots.

\[
f \not\equiv 0 \quad \Rightarrow \quad \Pr[f(r) = 0] \leq \frac{d}{|S|}
\]
Input: \( f \in \mathbb{F}[x_1, x_2, \ldots, x_n] \) of degree \( d \)

Output: \( f \equiv 0 ? \)

fix an arbitrary \( S \subseteq \mathbb{F} \)

pick random \( r_1, r_2, \ldots, r_n \in S \)

uniformly and independently at random;

check whether \( f(r_1, r_2, \ldots, r_n) = 0 \);

\[
f \equiv 0 \quad \Rightarrow \quad f(r_1, r_2, \ldots, r_n) = 0
\]

Schwartz-Zippel Theorem

\[
f \not\equiv 0 \quad \Rightarrow \quad \Pr[f(r_1, r_2, \ldots, r_n) = 0] \leq \frac{d}{|S|}
\]
Schwartz-Zippel Theorem

\[ f \neq 0 \quad \implies \quad \Pr[f(r_1, r_2, \ldots, r_n) = 0] \leq \frac{d}{|S|} \]

\[
f(x_1, x_2, \ldots, x_n) = \sum_{i_1, i_2, \ldots, i_n \geq 0 \atop i_1 + i_2 + \cdots + i_n \leq d} a_{i_1, i_2, \ldots, i_n} x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}
\]

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

\[
f(x_1, x_2, \ldots, x_n) = \sum_{i=0}^{d} x_n^i f_i(x_1, x_2, \ldots, x_{n-1})
\]

\[= g_{x_1, x_2, \ldots, x_{n-1}}(x_n)\]

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

\(g_{r_1, r_2, \ldots, r_{n-1}} \neq 0?\)
Schwartz-Zippel Theorem

\[ f \neq 0 \quad \implies \quad \Pr[f(r_1, r_2, \ldots, r_n) = 0] \leq \frac{d}{|S|} \]

**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

\[ f \neq 0 \quad \rightarrow \quad \Pr[ f(r_1, r_2, \ldots, r_n) = 0 ] \leq \frac{d}{|S|} \]

induction step:

**k:** highest power of \( x_n \) in \( f \)  \[ \left\{ \begin{array}{l} f_k \neq 0 \\ \text{degree of } f_k \leq d - k \end{array} \right\} \]

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

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

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

highest power of \( x_n \) in \( \bar{f} \)  \(< k \)
Schwartz-Zippel Theorem

\[ f \neq 0 \iff \Pr[f(r_1, r_2, \ldots, r_n) = 0] \leq \frac{d}{|S|} \]

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

\begin{align*}
\left\{ \begin{array}{l}
f_k \neq 0 \\
\text{degree of } f_k \leq d - k
\end{array} \right. \\
\text{highest power of } x_n \text{ in } \bar{f} < k
\end{align*}

law of total probability:

\[ \Pr[f(r_1, r_2, \ldots, r_n) = 0] \]

\[ = \Pr[f(\bar{r}) = 0 \mid f_k(r_1, \ldots, r_{n-1}) = 0] \cdot \Pr[f_k(r_1, \ldots, r_{n-1}) = 0] \]

\[ + \Pr[f(\bar{r}) = 0 \mid f_k(r_1, \ldots, r_{n-1}) \neq 0] \cdot \Pr[f_k(r_1, \ldots, r_{n-1}) \neq 0] \]

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

where \( g_{x_1, \ldots, x_{n-1}}(x_n) = f(x_1, \ldots, x_n) \)
Schwartz-Zippel Theorem

\[ f \not\equiv 0 \Rightarrow \Pr[f(r_1, r_2, \ldots, r_n) = 0] \leq \frac{d}{|S|} \]

\[ \Pr[f(r_1, r_2, \ldots, r_n) = 0] \leq \frac{d - k}{|S|} + \frac{k}{|S|} = \frac{d}{|S|} \]
Input: \( f \in \mathbb{F}[x_1, x_2, \ldots, x_n] \) of degree \( d \)

Output: \( f \equiv 0? \)

fix an arbitrary \( S \subseteq \mathbb{F} \)

pick random \( r_1, r_2, \ldots, r_n \in S \)

uniformly and independently at random;

check whether \( f(r_1, r_2, \ldots, r_n) = 0 \);

\[ f \equiv 0 \quad \Rightarrow \quad f(r_1, r_2, \ldots, r_n) = 0 \]

Schwartz-Zippel Theorem

\[ f \not\equiv 0 \quad \Rightarrow \quad \Pr [f(r_1, r_2, \ldots, r_n) = 0] \leq \frac{d}{|S|} \]
Thus, for any nondeterministic Turing machine $M$ that runs in some polynomial time $p(n)$, we can devise an algorithm that takes an input $w$ of length $n$ and produces $E_{M,w}$. The running time is $O(p^2(n))$ on a multitape deterministic Turing machine and...

WTF, man. I just wanted to learn how to program video games.