November 4, 2026

Today we'll start our study of randomness in algorithms & complexity. Randomized algorithms are often faster & simpler than deterministic algorithms. Does randomness add something essential, or can every randomized algorithm be simulated by a deterministic one?

We'll study this through the lens of the polynomial identity testing (PIT) problem.

Definition (Polynomial Identity Testing)
Input: arithmetic circuit C (over $\mathbb{C}$)
Output: YES if C computes zero poly
NO if C computes nonzero poly.

Remark. We can test equality of arithmetic circuits using PIT.

$$C(x) = D(x) \Leftrightarrow C(x) - D(x) = 0.$$

Why is PIT an interesting problem?

- It plays a fundamental role in randomized algebraic algorithms
- Close connections to circuit lower bounds
- Many surprising connections to problems outside of algebra.

There is a simple randomized algorithm based on the following lemma, called the Schwartz-Zippel lemma.

Lemma Let $f \in \mathbb{C}[x_1, \ldots, x_n]$ be nonzero. For any $S \subseteq \mathbb{C}$,

$$\Pr_{a \in S^n}[f(a) = 0] \leq \frac{\deg(f)}{|S|}.$$

Proof We proceed by induction on n.

$n = 1$: This is just the statement that a polynomial of degree d has $\leq d$ roots.

$n \geq 2$: Write

$$f(x_1, \ldots, x_n) = \sum_{i=0}^{e} x_n^i \cdot f_i(x_1, \ldots, x_{n-1}).$$

If $e = 0$, apply induction. Otherwise, we have $e > 0$. (Pick e such that $f_e \neq 0$.) Let

A = event $f(a_1, \ldots, a_n) = 0$

B = event $f_e(a_1, \ldots, a_{n-1}) = 0$

We have

$$\Pr[A] = \Pr[A \cap B] + \Pr[A \cap \bar{B}]$$

- $\Pr[A \cap B] \leq \Pr[B] \leq \frac{d-e}{|S|}$ by induction
- $\Pr[A \cap \bar{B}] = \Pr[A \mid \bar{B}] \Pr[\bar{B}]$

$$\leq \Pr[A \mid \bar{B}].$$

Conditioned on $\bar{B}$,

$$f(a_1, \ldots, a_{n-1}, x_n) = \sum_{i=1}^{e} x_n^i \, f_e(a_1, \ldots, a_{n-1})$$

is a nonzero polynomial of degree e. It is zero with probability $\leq e/|S|$. So

$$\Pr[A \mid \bar{B}] \leq \frac{e}{|S|}.$$

Putting things together gives us

$$\Pr[A] \leq \frac{d-e}{|S|} + \frac{e}{|S|} = \frac{d}{|S|} \qquad \square$$

We'll use Schwartz-Zippel to solve PIT for formulas.

<u>Theorem</u> There is a randomized, polynomial-time algorithm that takes as input an arithmetic formula and decides if the formula computes the zero polynomial. The algorithm errs with probability $1/100$.

Proof Let C be the size-s formula given as input.

Claim $\deg(C) \leq s$.

Proof Induction on formula size ☐

Let $S = \{1, 2, \ldots, 100s\}$.

By Schwartz-Zippel, if $C \neq 0$, then

$$\Pr_{a \in S^n}[C(a) = 0] \leq 1/100.$$

Algorithm:

1. Sample $a \in S^n$ uniformly at random
2. Evaluate $C(a)$
3. Return "$C(x) = 0$" iff $C(a) = 0$.

Complexity: formula eval is in poly time
Correctness: Schwartz-Zippel. ☐

Remark Same algorithm works for circuits that are promised to compute a polynomial of poly(n) degree.

For circuits of larger degree, we need to bound how large an evaluation can be.

Lemma If C is a constant-free size-s arithmetic circuit, then on input of absolute value $\leq B$, C evaluates to a number of absolute value $\leq B^{2^s}$.

Proof Induction on ckt size $\square$

Theorem PIT for arithmetic circuits can be solved in randomized polynomial time.

Proof Let C be a circuit of size s. Let $S = \{1, 2, \ldots, 100 \cdot 2^s\}$.

Schwartz-Zippel $\Rightarrow$ if $C \neq 0$, then

$$\Pr_{a \in S^n}[C(a) = 0] \leq 1/100.$$

By lemma, $|C(a)| \leq (100 \cdot 2^s)^{2^s}$.

This number has exponential bit complexity, too expensive to write down.

But it has at most

$$\log |C(a)| \leq 2^s \cdot \log(100 \cdot 2^s)$$
$$\leq O(2^s \cdot s)$$

distinct prime factors. If we randomly choose an s^2-bit prime p, then

$$\Pr[p \text{ divides } C(a)] \leq O\left(\frac{2^s \cdot s}{2^{s^2}/s^2}\right)$$

$$= O\left(\frac{2^s \cdot s^3}{2^{s^2}}\right) \ll 1/100.$$

Algorithm:

1) Sample $a \in S^n$
2) Sample s^2-bit prime p
3) Eval $C(a) \bmod p$
4) Report "$C=0$" iff $C(a) = 0$

Correctness: Schwartz-Zippel lemma + prime number theorem

Complexity: s arith ops, each on $\leq s^2$ bits $\Rightarrow$ poly(s) time

□

Next time: we'll look at derandomizing special cases of PIT.