Extended Church-Turing Thesis

• the belief that TMs formalize our intuitive notion of an efficient algorithm is:

The "extended" Church-Turing Thesis:

everything we can compute in time \( t(n) \)
on a physical computer can be computed on a Turing Machine in time
\( t(n)^{O(1)} \) (polynomial slowdown)

• randomized computation challenges this belief

Randomness in computation

• Example of the power of randomness

Randomized complexity classes

Communication complexity

Theorem: no deterministic protocol can compute \( \text{EQ}(x, y) \) while exchanging fewer than \( n+1 \) bits.

• Proof:
  • “input matrix”:

Communication complexity

• Can we do better?
  • deterministic protocol?
  • probabilistic protocol?
    • at each step: one party sends bits that are a function of held input and received bits so far and the result of some coin tosses
    • required to output \( f(x, y) \) with high probability over all coin tosses
Communication complexity

- protocol for EQ employing randomness?
  - Alice picks random prime $p$ in $\{1...4n^2\}$, sends:
    - $p$
    - $(x \mod p)$
  - Bob sends:
    - $(y \mod p)$
  - players output 1 if and only if:
    - $(x \mod p) = (y \mod p)$

- $O(\log n)$ bits exchanged
- if $x = y$, always correct
- if $x \neq y$, incorrect if and only if:
  - $p$ divides $|x - y|$,
- # primes in range is $\geq 2n$
- # primes dividing $|x - y|$ is $\leq n$
- probability incorrect $\leq 1/2$

Randomness gives an exponential advantage!!

Communication complexity

- Goal: compute $f(x, y)$ while communicating as few bits as possible between Alice and Bob
- Example: $EQ(x, y) = 1$ iff $x = y$
- Deterministic protocol: no fewer than $n+1$ bits
- Randomized protocol: $O(\log n)$ bits

Extended Church-Turing Thesis

- Common to insert “probabilistic”:
  - The "extended" Church-Turing Thesis:
  - everything we can compute in time $t(n)$ on a physical computer can be computed on a probabilistic Turing Machine in time $t(n)O(1)$ (polynomial slowdown)

Randomized complexity classes

- model: probabilistic Turing Machine
  - deterministic TM with additional read-only tape containing “coin flips”
  
  input tape: $01100111010000...$

  finite control: $q_0$

  read/write head: $01100111010000...$

  read head: $01100111010000...$

- $RP$ (Random Polynomial-time)
  - $L \in RP$ if there is a p.p.t. TM $M$:
    - $x \in L \rightarrow Pr_y[M(x,y) \text{ accepts}] \geq 1/2$
    - $x \notin L \rightarrow Pr_y[M(x,y) \text{ rejects}] = 1$

- $coRP$ (complement of Random Polynomial-time)
  - $L \in coRP$ if there is a p.p.t. TM $M$:
    - $x \in L \rightarrow Pr_y[M(x,y) \text{ accepts}] = 1$
    - $x \notin L \rightarrow Pr_y[M(x,y) \text{ rejects}] \geq 1/2$
  
  “p.p.t.” = probabilistic polynomial time
Randomized complexity classes

- **BPP** (Bounded-error Probabilistic Poly-time)
  - \( L \in \text{BPP} \) if there is a p.p.t. TM \( M \):
    - \( x \in L \rightarrow \Pr[M(x,y) \text{ accepts}] \geq 2/3 \)
    - \( x \notin L \rightarrow \Pr[M(x,y) \text{ rejects}] \geq 2/3 \)

One more important class:

- **ZPP** (Zero-error Probabilistic Poly-time)
  - \( ZPP = \text{RP} \cap \text{coRP} \)
  - \( \Pr[M(x,y) \text{ outputs "fail"}] \leq 1/2 \)
  - otherwise outputs correct answer

These classes may capture “efficiently computable” better than \( P \).

Relationship to other classes

- all these classes contain \( P \)
  - they can simply ignore the tape with coin flips
- all are in \( \text{PSPACE} \)
  - can exhaustively try all strings \( y \)
  - count accepts/rejects; compute probability
- \( \text{RP} \subseteq \text{NP} \) (and \( \text{coRP} \subseteq \text{coNP} \))
  - multitude of accepting computations
  - \( \text{NP} \) requires only one

Polynomial identity testing

- Given: polynomial \( p(x_1, x_2, \ldots, x_n) \) as arithmetic formula (fan-out 1):
  - multiplication (fan-in 2)
  - addition (fan-in 2)
  - negation (fan-in 1)

variables take values in finite field \( F \)

"polynomial identity testing" because given two polynomials \( p, q \), we can check the identity \( p \equiv q \) by checking if \( (p - q) \equiv 0 \)
Polynomial identity testing

• try all $|F|^n$ inputs?
  – may be exponentially many
• multiply out symbolically, check that all coefficients are zero?
  – may be exponentially many coefficients
• Best known deterministic algorithm places in $\text{EXP}$

Polynomial identity testing

Lemma (Schwartz-Zippel): Let $p(x_1, x_2, \ldots, x_n)$ be a total degree $d$ polynomial over a field $F$ and let $S$ be any subset of $F$. Then if $p$ is not identically 0,
\[
\Pr_{r_1, r_2, \ldots, r_n \in S}[ p(r_1, r_2, \ldots, r_n) = 0] \leq d / |S|.
\]

Polynomial identity testing

• Given: polynomial $p(x_1, x_2, \ldots, x_n)$ over field $F$
  \[ x_1 \quad x_2 \quad \cdots \quad x_n \]
• Is $p$ identically zero?
• Note: degree $d$ is at most the size of input

Randomized complexity classes

• We have shown:
  – Polynomial Identity Testing is in $\text{coRP}$
  – note: no sub-exponential time deterministic algorithm know

Randomized complexity classes

• How powerful is randomized computation?
• We have seen an example of a problem in $\text{BPP}$ that we only know how to solve deterministically in $\text{EXP}$.
  Is randomness a panacea for intractability?