Announcements:

• Q survey open
• Last Sections this week (through Friday)
Part I: Circuits: 
**Finite** computation, **quantitative** study

Part II: Automata: 
**Infinite restricted** computation, **quantitative** study

Part III: Turing Machines: 
**Infinite** computation, **qualitative** study

Part IV: Efficient Computation: 
**Infinite** computation, **quantitative** study

Part V: Randomized computation: 
Extending studies to non-classical algorithms
Randomized algorithm $ALG$ computes $F$ if for every input $x$

$$\Pr[ ALG(x) = F(x) ] \geq \frac{2}{3}$$

Probability over the randomness of the algorithm, not the input

The constant $2/3$ is arbitrary – can be replaced by $0.51$, $0.99$, even $1 - 2^{-n}$. Not by $1/2$.

BPP: {Boolean functions computable by some randomized algorithm}

Polynomial Identity Testing: in BPP, not known if in $P$.

$1/2$-approx to Max Cut: in BPP.
All functions $F: \{0,1\}^* \to \{0,1\}$

$R$ Computable functions
Today

- \( P \subseteq BPP \subseteq EXP \)
- \( BPP \subseteq P/poly \)
  - Proof uses success amplification via the Chernoff bond
- \( NP \) vs \( BPP \): Unknown, but Sipser-Gaacs-Lautemann Theorem:
  - If \( P = NP \) then \( BPP = P \)
  - \( BPP \) contained in a class like, and not much larger than, \( NP \).
Def 2: \( F: \{0,1\}^* \rightarrow \{0,1\} \) is in \( BPP \) if \( \exists \) poly-time deterministic algorithm \( A \), poly \( q(n) \) s.t. \( \forall n \forall x \in \{0,1\}^n \)

\[
\Pr_{r \sim \{0,1\}^{q(n)}}[A(x; r) = F(x)] \geq \frac{2}{3}
\]

\# possible \( q(n) \): \( 2^{q(n)} \leq 2^n \alpha n \)

Q: Prove that \( P \subseteq BPP \)  
A: Ignore randomness

Q: Prove that \( BPP \subseteq EXP \)  
A: Try all possible coin flip results
$BPP \subseteq P/poly$ outline in words

Def: $F: \{0,1\}^* \rightarrow \{0,1\}$ is in $BPP$ if $\exists$ poly-time deterministic algorithm $A$ such that $\forall n$, given a random poly-size advice string $q(n)$, $\forall x \in \{0,1\}^n$, $A$ decides $F(x)$ right, $p > 2/3$.

Def: $F: \{0,1\}^* \rightarrow \{0,1\}$ is in $P/poly$ if $\exists$ poly-time deterministic algorithm $A$ such that $\forall n$, given a fixed poly-size advice string $q(n)$, $\forall x \in \{0,1\}^n$, $A$ decides $F(x)$ right.

Proof idea: Amplify the success probability so much that one $q(n)$ works for every input.
\(BPP \subseteq P/poly\) outline in pictures

\[\Pr_{x \sim \{0,1\}^n, r \sim \{0,1\}^m} [A(x; r) = F(x)] \geq \frac{2}{3}\]

\[\forall x \in \{0,1\}^n \Pr_{r \sim \{0,1\}^m} [A(x; r) = F(x)] \geq \frac{2}{3}\]
Amplification for 2-sided error

Thm: If $F \in BPP$ then $\exists$ poly-time algorithm $B$, poly $q(n)$ s.t. $\forall n \forall x \in \{0,1\}^n$

$$\Pr_{r \sim \{0,1\}^{q(n)}}[B(x;r) = F(x)] \geq 1 - 2^{-n^2}$$

Generally: Can amplify success from $\frac{1}{2} + \frac{1}{p(n)}$ to $1 - 2^{-r(n)}$ for all polys $p, r$

Chernoff Bound: Let $X_0, \ldots, X_{n-1}$ i.i.d. r.v.’s with $X_i \in [0,1]$. Then if $X = X_0 + \cdots + X_{n-1}$ and $p = \mathbb{E}[X]$, for every $\epsilon > 0$,

$$\Pr[|X - np| > \epsilon n] < 2^{1-(2 \lg e)\epsilon^2 n}$$
Thm: If $F \in BPP$ then $\exists$ poly-time algorithm $B$, poly $q(n)$ s.t. $\forall n \ \forall x \in \{0,1\}^n$,

$$\Pr_{r \sim \{0,1\}^{q(n)}}[B(x; r) = F(x)] \geq 1 - 2^{-n^2}$$

Proof: Suppose $\Pr[A(x; r) = F(x)] \geq 2/3$.

Idea: $B$ will run $A\ 1000n^2$ times and return majority vote.

Define $X_i = \begin{cases} 1, & A(x; r_i) = F(x) \\ 0, & A(x; r_i) \neq F(x) \end{cases}$

$X_1, ..., X_{1000n^2}$ i.i.d with $\mathbb{E}[X_i] \geq 2/3$

By Chernoff, $\Pr[\frac{1}{1000n^2} \sum_i X_i < 0.5] < 2^{1 - \frac{2 \lg e}{36} \cdot 1000n^2} < 2^{-n^2}$
If \( F \in BPP \) then by amplification \( \exists \) poly time algorithm \( A \) s.t.
\[
\Pr_{r \sim \{0,1\}^m} [A(x; r) \neq F(x)] < 2^{-n}
\]

Let \( M = 2^m \) be # of random choices

Let \( N = 2^n \) be # of inputs

Every column has \( < \frac{M}{N} \) “reds”

\[ \Rightarrow \text{# reds} < \text{# rows} \]
\[ \Rightarrow \text{must be rows with no reds!} \]

A “good choice of randomness” \( r^* \) s.t.
\[ \forall x \in \{0,1\}^n \ A(x; r^*) = F(x) \]

Use \( r^* \) as the P/poly advice string.
**Proof:** Suppose \( F \in BPP \) and \( A \) is alg using \( n^a \) random bits and running in \( n^b \) time s.t. \( \Pr[A(x;r) \neq F(x)] < 0.001 \cdot 2^{-n} \).

By \( P \subseteq P/poly \) there’s circuit \( C \) of \( \leq n^{4b} \) computing \( x, r \mapsto A(x; r) \).
\( \text{BPP} \subseteq \text{P/poly} \) denouement

Proof: Suppose \( F \in \text{BPP} \) and \( A \) is alg using \( n^a \) random bits and running in \( n^b \) time s.t. \( \Pr[ A(x;r) \neq F(x) ] < 0.001 \cdot 2^{-n} \)

By \( \text{P} \subseteq \text{P/poly} \) there’s circuit \( C \) of \( \leq n^{4b} \)
computing \( x, r \mapsto A(x;r) \)

\[ n \quad \begin{array}{c} x \end{array} \quad \begin{array}{c} C \end{array} \quad \begin{array}{c} n^a \quad \begin{array}{c} r \end{array} \end{array} \]

\( \leq n^{2b} \) gates

\[ M = 2^n \] possible random choices

\[ N = 2^n \] possible inputs
\( BPP \subseteq P/poly \) deenouement

**Proof:** Suppose \( F \in BPP \) and \( A \) is alg using \( n^a \) random bits and running in \( n^b \) time s.t. \( \Pr[A(x; r) \neq F(x)] < 0.001 \cdot 2^{-n} \)

By \( P \subseteq P/poly \) there’s circuit \( C \) of \( \leq n^{4b} \)

computing \( x, r \mapsto A(x; r) \)

\( x \mapsto A(x; r^*) \) is the map \( F \) on \( \{0,1\}^n \)!
Recap for now

- $P \subseteq BPP$
- $BPP \subseteq EXP$
- $BPP \subseteq P/poly$
- Unknown if $BPP = P$. Unknown if $BPP = EXP$

Q: Can it be that $P = BPP = EXP$? 

Q: Is there a poly-time deterministic algorithm that given randomized alg $A$ for $F \in BPP$ and $n \in \mathbb{N}$ outputs a circuit $C_n$ that computes $F$ on $\{0,1\}^n$?
**BPP and NP**

**Q:** Suppose that $F \leq_p G$ and $G \in BPP$. Prove that $F \in BPP$.

$3SAT \in \mathsf{P}$ \hspace{1cm} $NP \subseteq \mathsf{P}$

**Corollary:** If $3SAT \in BPP$ then $NP \subseteq BPP$

**Unknown:** Is $BPP \subseteq NP$? Is $NP \subseteq BPP$? Both? Neither?

**Known:** Sipser-Gaacs-Lautemann Theorem: If $P = NP$ then $BPP = P$
**Sipser-Gaacs-Lautemann Thm:** If $P = NP$ then $BPP = P$

**Proof idea:** First, amplify like crazy:

Ensure: $\Pr_{r \sim \{0,1\}^m}[A(x; r) = F(x)] \geq 1 - 2^{-n} > 1 - \frac{1}{1000m}$

\[ S_x := \{r | A(x; r) = 1\} \]

$F(x) = 0 : |S_x| < \frac{1}{1000m} 2^m$

$F(x) = 1: |S_x| > \left(1 - \frac{1}{1000m}\right) 2^m$

**MAIN LEMMA:** $F(x) = 1$ iff $\exists m$ shifts $s_1, \ldots, s_m$ s.t. $\{0,1\}^m = \cup (S_x \oplus s_i)$

$F(x) = 1$ iff $\exists s_1, \ldots, s_m \in \{0,1\}^m \forall z \in \{0,1\}^m \exists i \in [m] \exists r \in \{0,1\}^m : (A(x; r) = 1) \land (z = r \oplus s_i)$
Ensure: $\Pr_{r \sim \{0, 1\}^m}[A(x; r) = F(x)] \geq 1 - 2^{-n} \geq 1 - \frac{1}{1000m}$

$S_x := \{r | A(x; r) = 1\}$

- $A(x; r) = 0$
- $A(x; r) = 1$

$F(x) = 0 : |S_x| < \frac{1}{1000m} 2^m$
$F(x) = 1 : |S_x| > \left(1 - \frac{1}{1000m}\right) 2^m$

**MAIN LEMMA:** $F(x) = 1$ iff $\exists m$ shifts $s_1, \ldots, s_m$ s.t. $\{0, 1\}^m = \bigcup (S_x \oplus s_i)$

**CLAIM 1 ($\Leftrightarrow$):** If $|S| < \frac{1}{1000m} 2^m$ then $\forall s_1, \ldots, s_m \in \{0, 1\}^m$ $|\bigcup_i (S \oplus s_i)| < 2^m$

**Proof:** $|S \oplus a| = |S|$

$|\bigcup_i (S \oplus s_i)| < m \cdot \frac{1}{1000m} 2^m < 2^m$
Ensure: \( \Pr_{r \sim \{0,1\}^m} [A(x; r) = F(x)] \geq 1 - 2^{-n} \geq 1 - \frac{1}{1000m} \)

\( S_x := \{ r | A(x; r) = 1 \} \)

\( A(x; r) = 0 \)

\( A(x; r) = 1 \)

\( F(x) = 0 : |S_x| < \frac{1}{1000m} 2^m \)

\( F(x) = 1: |S_x| > \left(1 - \frac{1}{1000m}\right) 2^m \)

**MAIN LEMA:** \( F(x) = 1 \) iff \( \exists m \) shifts \( s_1, ..., s_m \) s.t. \( \{0,1\}^m = \cup (S_x \oplus s_i) \)

**CLAIM 2 \( (\Rightarrow) \):** If \( |S| > \frac{2}{3} 2^m \) then \( \exists s_1, ..., s_m \in \{0,1\}^m \) s.t. \( \cup_i (S \oplus s_i) = \{0,1\}^m \)

**Proof:** For every \( z \in \{0,1\}^m \)

\( \Pr_s [z \notin S \oplus s] = \Pr_s [s \notin S \oplus z] < \frac{1}{3} \)

\( \Rightarrow \) For every \( z \in \{0,1\}^m \)

\( \Pr_{s_1, ..., s_m} \left[ \bigwedge_{i=1}^m z \notin S \oplus s_i \right] < \left(\frac{1}{3}\right)^m < 2^{-m} \)

\( \Rightarrow \) \( \Pr_{s_1, ..., s_m} \left[ \exists z \in \{0,1\}^m \bigwedge_{i=1}^m z \notin S \oplus s_i \right] < 1 \) \( \Rightarrow \exists s_1, ..., s_m \)
**Sipser-Gaacs-Lautemann Thm:** If $P = NP$ then $BPP = P$

**MAIN LEMMA:** $F(x) = 1$ iff $\exists m$ shifts $s_1, \ldots, s_m$ s.t. $\{0,1\}^m = \cup (S_x \oplus s_i)$

\[
F(x) = 1 \text{ iff } \exists_{s_1,\ldots,s_m \in \{0,1\}^m} \forall_{z \in \{0,1\}^m} \exists_{i \in [m]} \exists_{r \in \{0,1\}^m} : (A(x; r) = 1) \land (z = r \oplus s_i)
\]

\[
F(x) = 1 \text{ iff } \exists_{s_1,\ldots,s_m} \exists_{z} \exists_{i \exists_r} : (A(x; r) = 1) \land (z = r \oplus s_i)
\]

In NP, so replace with P alg (no $\exists_i \exists_r$)

Also in P

In NP, so replace with P alg (no $\exists_z$)

Also in P

In NP, so replace with P alg (no $\exists_{s_1,\ldots,s_m}$)

$NP = \Sigma_1^p$

$S_2$

$NP = \Sigma_3$

$f(s_1,\ldots,s_m, z) = 1$ iff $\exists i \exists r$ ....

$f(s_1,\ldots,s_m) = 1$ iff $\exists z$ ....
**BPP and NP recap**

- If $3SAT \in BPP$ then $NP \subseteq BPP$: All theory of $NP$ completeness stays the same if we use $BPP$ as our model of “efficient computation”.

- If $P = NP$ then $BPP = P$

- If (as widely believed) $3SAT \notin P_{/poly}$ then $NP \not\subseteq BPP$
All functions $F : \{0,1\}^* \rightarrow \{0,1\}$

$R$ Computable functions

- $P$
- $BPP = EXP$
- $NP$
- $NPC$
- $2SAT$
- $3SAT$

Unknown but believed false

$HALT_{2^n}$

$HALT$

$UHALT$
All functions $F: \{0,1\}^* \rightarrow \{0,1\}$

- $R$: Computable functions
- $P$: Functions computable in polynomial time
- $P/poly$: Functions computable in polynomial time with advice
- $EXP$: Exponential time functions
- $NP$: Functions in the class of non-deterministic polynomial time
- $NPC$: Functions in the class of NP-complete
- $2SAT$, $PRIMALITY$, $3SAT$, $FACTORIZATION$, $PIT$: Specific problems
- $HALT_{2^n}$, $UHALT$: Halting problems

Unknown but believed to be true
BPP recap

- $P \subseteq BPP \subseteq EXP$
  Unknown if either inclusion strict but can’t have $P = BPP = EXP$
- $BPP \subseteq P/poly$
- If $BPP$ contains an $NP$-complete problem then $NP \subseteq BPP$
- Relation with $NP$ unknown
- If $P = NP$ then $BPP = P$
- It is believed that $P \neq NP$ (of course) but it is also believed that $BPP = P$. 
Next Lecture: Wrap-up

• Quantum Computation
  • Most credible challenger to Strong Church-Turing Thesis.
  • Less-Strong Church-Turing Thesis: SCTT but with quantum computers.

• Cryptography, Society

• Exam info
Bonus topics:

- One sided error algorithms: $coRP, RP$
- "Zero sided error" (Las Vegas): $ZPP$
- Known that $ZPP = RP \cap coRP$ and that $RP \cup coRP \subseteq BPP$
- Known that $RP \subseteq NP$ and $coRP \subseteq coNP$
- Pseudorandom generators
- relation between counting and sampling.
- Randomized reductions.