Example Problems

- Given $n$-bit integer $N$, find a prime $p \in [N + 1, N^2]$.
- Given $n$-bit prime $p$ and integer $a$, find $\alpha$ such that $\alpha^2 = a \pmod{N}$.
- Given $k$, $n \times n$ matrices $M_1, \ldots, M_k$ over the integers, do there exist integers $\gamma_1, \ldots, \gamma_k$ such that $\sum_{i=1}^{k} \gamma_i \cdot M_i$ is nonsingular.
- Given algebraic circuits $C_1$ and $C_2$ over integers, are the two circuits computing the same function?

Why so common? Some basic tools

- If $H$ is a subgroup of finite group $G$, then $|H|/|G| \leq 1/2$.
- # Primes in $[1, N]$ is $N/\ln N(1 + o(1))$.
- If $N \leq 2^n$ and $p_1, \ldots, p_{2n}$ are relatively prime, then $N \equiv 0 \pmod{p_i}$ for at most $n$ of the $p_i$'s.
- Multivariate polynomial $p$ of degree $d$ in $n$ variables vanishes w.p. at most $d/|S|^n$ over a random element of $S^n$. (Prove the last?)

Randomization

- Example of randomized algorithms

Model, Classes

Detour: Promise problems

Upcoming results:
- Amplification of RP/BPP.
- BPP in P/poly.
- BPP in PH.

Physicists' Belief: Natural phenomena have randomness built into them.

How does this affect our belief that "polynomial time" is all that is feasible?

Should study formally.

Are there examples of computations that are performed efficiently with randomization, but not without? (Yes! Several in number theory/algebra.)


Example application: Square roots

- Finding square root of $a \mod p$.
- Equivalent to factoring $x^2 - a$ in $\mathbb{Z}_p[x]$.
- Key idea: $(x^2 - a) = (x - \alpha)(x + \alpha)$ which divides $\prod_{\beta \in \mathbb{Z}_p}(x - \beta) = x^p - x$.
- Factor right hand side into $q_1(x) \cdot q_2(x)$ and hope $\gcd(x^2 - a, q_1(x))$ is non-trivial.

- One factorization of RHS: $x^p - x = x(x^{(p-1)/2} - 1)(x^{(p-1)/2} + 1)$. Hope $x - \alpha|x^{(p-1)/2} - 1$ but $x + \alpha$ doesn’t. I.e., $\alpha^{(p-1)/2} = 1$ but $(-\alpha)^{(p-1)/2} = -1$. Happens only if $(-1)^{(p-1)/2} = -1$. Which happens only if $p = 3(\mod 4)$. What about the other cases?

Models: Randomized algorithms/Turing machines

- Model 1: Machine can enter a random state whenever it wishes. Takes one of two outgoing transitions randomly.
- (Equivalent) Model 2: Machine has two inputs: (1) The actual input and (2) the outcome of many independent random coin tosses.

Randomized machines and languages

Machine $M$ for Language $L$ has:

**Completeness** $c$ if $c = \inf_{x \in L} \Pr_y[M(x, y)\text{accepts}]$ (Assume uniform distribution on $\ell(|x|)$ bit strings.

**Soundness** $s$ if $s = \sup_{x \notin L} \Pr_y[M(x, y)\text{accepts}]$.

$M$ seems to decide membership in $L$ if $c > s$. But even better if $c = 1$ (and/or $s = 0$).
- Resource? Space or Time?

- What kind of error? Two attributes; Four classes.
  - "False positives": Says $x \in L$ while $x \notin L$. (Soundness $> 0$.)
  - "False negatives": Says $x \notin L$ when $x \in L$. (Completeness $< 1$.)

- All in all, get eight classes!

---

**Time-bounded randomization**

- **BPP**: (Bounded Probability Polynomial-time): Both kinds of errors allowed (two-sided error): $L \in \textit{BPP}$ if there exists a two-input deterministic machine $M$ running in time poly in first input such that:

$$x \in L \Leftrightarrow \Pr_{y}[M(x,y)\text{ accepts}] \geq 2/3.$$  

(Completeness $= 2/3$; Soundness $= 1/3$).

- **RP**: (Randomized Polynomial-time): Only false negatives (one-sided error):

$$x \in L \Rightarrow \Pr_{y}[M(x,y)\text{ accepts}] \geq 2/3.$$  

(Completeness $= 2/3$; Soundness $= 0$ (perfect)).

---

**Space-bounded randomization**

Similar collection of four classes:

- **BPL, RL, co-RL, ZPL**.

- Catch 1: In two-input model, have one way access to second input.

- Catch 2: Machines bounded to run in polynomial time.

---

**Time-bounded randomization (contd.)**

- **co-RP**: complements of RP languages.

- **ZPP**: Error happens with probability zero! So what does randomness do? Running time is not guaranteed to be polynomial. Only expected to be polytime.
• 2/3, 1/3 arbitrarily chosen. For definition of BPP suffices to have \( c > s \). Similarly for RP, suffices to have \( c > 0 \) etc.

• Randomness more powerful than deterministic?
  – Belief: No.
  – Current evidence: Yes. There exist problems in RP that we can show to be in P. (Example: Primality testing.) There exist problems in RL that we can’t show to be in L. (Example: USTCON - connectivity in undirected graphs.)

• How do RP, BPP etc. relate to familiar complexity classes.
  – Obviously: ZPP in RP & co-RP; and all are in BPP.
  – RP in NP (by definition).
  – BPP? Don’t quite know:
    – BPP in \( P^{/\text{poly}} \).
    – BPP in PH.