Derandomization

- Power of Randomness
    - Identity Testing
    - Probabilistic Method
    - “Finding Hay in a Haystack”
    - Approximation Algorithms for NP Hard Problems
        - Max Cut and MAX 3-SAT CNF
- Quick Review of Properties of Expectations
    - Linearity of Expectations
    - Method of Iterated Expectations
    - Markov’s Inequality
        - Chernoff Bounds
- Review 3-CNF MAX SAT Approximation Algorithm
    - Expectation of Random Guessing is 7k/8
    - Algorithm
        - Guess Randomly
        - Stop when you satisfy >= 7k/8
    - Expected Runtime:

      Proof.

      Let p_j be probability that exactly j clauses are satisfied; let p be probability that $\geq 7k/8$ clauses are satisfied.

      $$\begin{aligned} \frac{7}{8}k = E[Z] = \sum_{0 \leq j \leq k} j\,p_j &= \sum_{0 \leq j < 7k/8} j\,p_j + \sum_{7k/8 \leq j \leq k} j\,p_j \\ &\leq \left(\frac{7}{8}k - \frac{1}{8}\right) \sum_{0 \leq j < 7k/8} p_j + k \sum_{7k/8 \leq j \leq k} p_j \\ &\leq \left(\frac{7}{8}k - \frac{1}{8}\right) \cdot 1 + kp \end{aligned}$$

      Solving for p yields $p \geq 1/(8k)$. □
    - Run time is at most 8k^2
- Derandomizing 3-CNF MAX SAT
    - Raw Enumeration
    - Method of Conditional Expectations

        - Pessimistic Estimators
- Pairwise Independence
    - Strongly 2-Universal Hash Functions
    - 2-Universal Hash Functions
- Complexity Results and Questions:
    - RP (Randomized Polynomial Time)
        - Always runs in Polynomial Time
        - If not in the Language always output 0
        - If in Language output 1 with probability at least ½
    - BPP (Bounded Error Probabilistic Polynomial Time)
        - Always run is polynomial time for all inputs
        - If not in the language output 1 with probability at most ⅓
        - If in the language output 1 with probability at least ⅔
    - RP \subset BPP \subset EXP
        - EXP just takes in with the extra random bits
    - P \subset RP \subset NP
        - Certificate is the coin flips
    - P = RP?
    - BPP \subseteq NP?
    - P = BPP?
        - Can we always use O(log(n)) random bits?
            - Possible approach to proving this.
        - AKS was in BPP for a while until 2002 when AKS moved it to P.