COMPSCI 614: Problem Set 1

Due: 2/21 by 11:59pm in Gradescope.

Instructions:

- You are allowed to, and highly encouraged to, work on this problem set in a group of up to three members.
- Each group should **submit a single solution set**: one member should upload a pdf to Gradescope, marking the other members as part of their group in Gradescope.
- You may talk to members of other groups at a high level about the problems but **not work through the solutions in detail together**.
- You must show your work/derive any answers as part of the solutions to receive full credit.

Hint: The following two inequalities may be helpful in various places (and generally throughout the course): for any $x > 0$, $(1+x)^{1/x} \leq e$ and $(1-x)^{1/x} \leq 1/e$.

1. Randomized Complexity Classes (12 points)

1. (2 points) Suppose I have a randomized algorithm $\mathcal{A}$ that is always correct on 'NO' instances and succeeds with probability at least 1/2 on 'YES' instances. Describe and analyze algorithm that makes a *single call* to $\mathcal{A}$ and succeeds with probability at least 2/3 on both 'YES' and 'NO' instances. Explain in a sentence or two why this shows that $RP \subseteq BPP$.
2. (2 points) Suppose I have a Las Vegas algorithm that solves a decision problem with expected runtime T. For any $\delta > 0$, describe and analyze a Monte-Carlo algorithm that correctly answers the decision problem with probability at least $1-\delta$ and has worst case runtime $O(\log(1/\delta) \cdot T)$. Explain in a sentence or two why this result shows that $ZPP \subseteq BPP$. **Hint:** It is possible to solve this problem justing using Markov's inequality and basic probability computations.
3. (2 points) Suppose I have a Monte-Carlo algorithm that solves a decision problem with worst case runtime T and at least 2/3 probability of correctness. For any $\delta > 0$, describe and analyze a Monte-Carlo algorithm that correctly answers the decision problem with probability at least $1-\delta$ and has worst case runtime $O(\log(1/\delta) \cdot T)$.
4. (2 points) Consider the set of problems solvable by a randomized algorithm that runs in polynomial time and outputs correctly with probability at least 3/4 on 'YES' instances, and probability at least 1/4 on 'NO' instances. Is this an interesting complexity class? Is it equivalent to any of the classes discussed in Lecture 1? What about if the algorithm must be correct with probability 4/5 on 'YES' instances and probability 1/4 on 'NO' instances?

5. (2 points) Consider the set of problems solvable by a randomized algorithm that runs in polynomial time and is correct with probability at least $1/2 + 1/n$ on any input instance. Is this an interesting complexity class? Is it equivalent to any of the classes discussed in Lecture 1?

6. (2 points) Prove that 3-SAT is in PP, and thus that $NP \subseteq PP$. **Note:** You can easily find the solution to this question online. Try to come up with the answer yourself.

2. Polynomial Identity Testing (6 points)

1. (2 points) In class we proved the Schwartz-Zippel Lemma: if one picks $(z_1, \ldots, z_n)$ uniformly at random from S^n (i.e., from the set of n-length vectors with entries lying in the set S) then for any n-variable non-zero polynomial p with degree d, $\Pr[p(z_1, \ldots, z_n) = 0] \leq \frac{d}{|S|}$. Describe, for all values of d and n, an example where this bound is tight. I.e., where $\Pr[p(z_1, \ldots, z_n) = 0] = \frac{d}{|S|}$.

2. (4 points) Say I give you three matrices $A, B, C \in \mathbb{R}^{n \times n}$ and I want you to determine if $AB = C$. You can do this in $O(n^3)$ time deterministically (or actually $O(n^\omega)$ time for $\omega \approx 2.37$) by just computing AB and checking equality with C. But you can do it much faster with a randomized algorithm: Let $x \in \mathbb{R}^n$ be a random vector. Check if $ABx = Cx$. If yes, return that $AB = C$. Otherwise return that $AB \neq C$.

   (a) (1 point) What is the runtime of this algorithm? **Note:** Assume that basic arithmetic operations like addition and multiplication of two numbers take $O(1)$ time.

   (b) (1 point) What is the probability that the algorithm is correct when $AB = C$?

   (c) (2 points) Prove that if x has entries picked independently and uniformly at random from the integers $\{1, 2, \ldots, s\}$, then this algorithm is correct with probability at least $1 - 1/s$ in the case that $AB \neq C$. **Hint:** View $ABx - Cx$ as a polynomial in x and apply the Schwartz-Zippel Lemma.

3. Finding Random Primes (4 points)

A common application of randomized primality testing algorithms is in finding large random primes for use in cryptographic schemes, such as RSA. Suppose you have a Monte-Carlo primality testing algorithm that, given an integer input x, outputs 'yes' or 'no'. If x is prime, the algorithm always outputs 'yes'. If x is composite, the algorithm outputs 'no' with probability at least $1/2$.

For any $\delta \geq 0$, describe an algorithm that makes $O(\log(n) \cdot \log(1/\delta))$ calls to this tester and outputs x such that, with probability at least $1 - \delta$, x is a uniformly random prime in $\{1, \ldots, n\}$.

Hint 1: Let $\pi(n)$ be the *prime counting function*: the number of prime numbers less than or equal to any integer n. You may use the fact that for any $n \geq 17$, $\frac{n}{\log n} \leq \pi(n)$. This fact is closely related to the prime number theorem (PNT).

Hint 2: It might be easier to first shoot for $O(\log(n) \cdot \log(1/\delta) \cdot \log(\log n/\delta))$ calls or something like this and then figure out how to tighten your analysis. You will get partial credit for any weaker bound that just loses log factors in n and $1/\delta$.

To think about: Why in cryptographic applications do we want a random prime in $\{1, \ldots, n\}$ rather than say, any prime at least as big as $n/2$?

4. Tighter Bound for Coupon Collecting (3 points)

Analyze the coupon collecting problem by considering a union bound over the events $E_1, \ldots, E_n$ where E_i is the event that you do not collect coupon i in T rounds. Prove that for $T = cn \ln n$, with probability at least $1 - \frac{1}{n^{c-O(1)}}$ you collect all coupons within T rounds. How does this compare to the bound that is given by the variance analysis + Chebyshev's inequality shown in class?

5. Randomized Routing (4 points + 3 bonus points)

Consider the following simplified model of a routing problem under communication constraints: we have n nodes in a fully connected network. Each node has n messages that it would like to deliver to n (not necessarily unique) destinations. Also, for simplicity, we assume that each node is the intended recipient of exactly n messages. In each round, a node can send at most one message along each connection in the network. Assume that the message contains the id of its final intended recipient. Naively, sending these messages could take n rounds, if e.g., some node v needs to send n messages to some other node u.

Describe and analyze a randomized scheme that sends all messages in $O(\log n)$ rounds with high probability, i.e., with probability at least $1 - 1/n^c$ for a large constant c.

Hint: Have each node initially send each of its n messages to a random recipient, who will then forward the message to its final destination.

3 Bonus Points (Quite Challenging): Can you achieve $O(\log \log n)$ rounds?