6.842 Randomness and Computation February 7, 2022

Homework 1

Lecturer: Ronitt Rubinfeld *Due Date: February 23, 2022*

Homework guidelines: You may work with other students, as long as (1) they have not yet solved the problem, (2) you write down the names of all other students with which you discussed the problem, and (3) you write up the solution on your own. No points will be deducted, no matter how many people you talk to, as long as you are honest. If you already knew the answer to one of the problems (call these "famous" problems), then let us know that in your solution writeup – it will not affect your score, but will help us in the future. It's ok to look up famous sums and inequalities that help you to solve the problem, but don't look up an entire solution.

The following problems are to help you understand the upcoming lectures. Please make sure you can do them. (Hopefully, some of them will be fun to think about). Do not turn them in.

1. (This is the "Von Neumann trick", which you don't really need for upcoming lectures, but it's cute). Given a coin with probability p of getting "heads", give a procedure for simulating one toss of a fair coin ($p = 1/2$). The procedure should run in expected time that is polynomial in $\frac{1}{p} + \frac{1}{1-p}$.
2. (The following uses a very important technique that we will make use of extensively throughout the course). You are given $n \times n$ matrices A, B, C whose elements are from $\mathbb{Z}_2$ (integers mod 2). Show a (randomized) algorithm running in $O(n^2)$ time which verifies $A \cdot B = C$. The algorithm should always output "pass" if $A \cdot B = C$ and should output "fail" with probability at least 3/4 if $A \cdot B \neq C$. Assume the field operations $+, \times, -$ can be done in $O(1)$ steps.
3. A 3-SAT formula takes the "and" of a set of clauses, where each clause takes the "or" of a set of literals (each literal is a variable, or the negation of a variable). Show that for any 3-SAT formula in which every clause contains literals corresponding to 3 distinct variables, there is an assignment that satisfies at least 7/8 of the clauses.

The following problems are to be turned in.

1. You are given an approximation scheme $\mathcal{A}$ for f such that $Pr[\frac{f(x)}{1+\epsilon} \leq \mathcal{A}(x) \leq f(x)(1+\epsilon)] \geq 3/4$, and $\mathcal{A}$ runs in time polynomial in $1/\epsilon, |x|$. Construct an approximation scheme $\mathcal{B}$ for f such that $Pr[\frac{f(x)}{1+\epsilon} \leq \mathcal{B}(x) \leq f(x)(1+\epsilon)] \geq 1 - \delta$, and $\mathcal{B}$ runs in time polynomial in $\frac{1}{\epsilon}, |x|, \log \frac{1}{\delta}$.
2. Denote the complete graph on n nodes by K_n. Let $R(t)$ be the minimal n such that for any two-coloring of the edges of K_n, there is a subset of the vertices of K_n, of size t, such that all edges between vertices in this subset are the same color.

   Show that if $\binom{m}{t}2^{1-\binom{t}{2}} < 1$ then $R(t) > m$. (i.e., show that if $\binom{m}{t}2^{1-\binom{t}{2}} < 1$, then there is a coloring such that there is no subset of vertices of size t such that the edges joining these vertices are all one color).

3. Given a Boolean function $f(\cdot)$ on Boolean inputs, a sequence $C = C_1, C_2, ...$ of circuits is a *circuit family for* $f(\cdot)$ if C_n has n input bits $(x_1, \ldots, x_n)$, and computes $f(x_1, \ldots, x_n)$ as its output bit. The family C is said to be *polynomial-sized* if the size of C_n (number of "and", "or" and "not" gates) is bounded above by $p(n)$ for every n, where $p(\cdot)$ is a polynomial. A *randomized circuit family for* $f(\cdot)$ is a circuit family for $f(\cdot)$ that, in addition to the n inputs $x_1, \ldots, x_n$, takes m random input bits $r_1, \ldots, r_m$, each of which is equiprobably and independently 0 or 1. In addition, for every n, circuit C_n must satisfy

   (a) if $f(x_1, \ldots, x_n) = 0$ then output 0 regardless of the values of the random inputs $r_1, \ldots, r_m$.

   (b) if $f(x_1, \ldots, x_n) = 1$ then output 1 with probability $\geq 1/2$ over the choice of $r_1, \ldots, r_m$.

   Show: If a Boolean function can be computed by a randomized polynomial sized circuit family, then it can be computed by a deterministic polynomial sized circuit family.

   Hint: It is useful to think of the circuit as an algorithm that works only for inputs of a specific size n. You need to show that if there is a circuit that correctly computes for all inputs of size n using random bits, then there is a polynomial sized circuit that correctly computes for all inputs of size n without the help of random bits. To do this, find a single random string that works for all inputs of size n and encode it into the circuit. This is probably not possible for the original randomized circuit as it is given, so you will have to first "improve" the circuit.

4. **(Directed cycles)** Let $D = (V, E)$ be a simple directed graph (that is, a directed graph with no self-loops and with at most one edge between ever pair of vertices). Assume that D has minimum outdegree δ and maximum indegree Δ. Show that if $e(\Delta\delta+1)(1-\frac{1}{k})^{\delta} < 1$, then D contains a (directed, simple) cycle whose length is a multiple of k.
   Hint: Let $f : V \to \{0, 1, \ldots, k-1\}$ be a random coloring of V obtained by choosing for each $v \in V$, $f(v) \in \{0, \ldots, k-1\}$ independently according to a uniform distribution. For each $v \in V$, consider the event A_v that there is no $u \in V$ s.t. $(v, u) \in E$ and $f(u) = f(v) + 1$ mod k.