COMP 531: Advanced Theory of Computation (Winter 2014)

Assignment 4

Due March 31st

Instructions

Follow these instructions closely.

You will benefit most if you seriously try solving each problem yourself. You may work with each other but you must write up your own solutions. For each question, you should clearly acknowledge the people you have worked with. You are not allowed to use any resources that contain the solution to an assignment question. However, we value honesty above all. You will get full marks if you happen to find the solution to a question and you write your own solution, **as long as you properly acknowledge your source**. Failure to acknowledge your source can result in 0 points.

Clarity and conciseness of your solutions are as important as correctness. It is important to learn how to write your ideas and solutions clearly and rigorously. You will lose marks for correct solutions that are poorly explained/presented. When writing your solutions, assume that your audience is your class mates rather than the instructor of the course. The high level ideas and an overview of your argument should be presented before any technical details, and all non-trivial claims have to be proven.

If you do not know how to solve a problem, do not answer it. This will earn you 20% of the points. Do not make yourself believe in a wrong proof, this is bad for you. **And definitely do not try to sell it!** If you don't know how to solve a problem but you have some non-trivial ideas, write them down. If you have a solution with gaps, write your argument and clearly indicate the gaps.

Submit your assignments in class or send a copy to `aada@cs.mcgill.ca` before midnight of the due date.

Questions

1. The *probabilistic method* is a powerful technique to prove the existence of objects with desired combinatorial properties. The idea is to define a suitable probability distribution in which the probability of finding the desired properties is non-zero. As an illustration of the method, we do the following exercise:

   Let G be any graph that has a matching M (a matching is a collection of edges no two of which intersect each other at any vertex). A subgraph H of G is any graph whose set of vertices and edges is a subset of the set of vertices and edges of G.

   Show that G contains a subgraph H, where H is bipartite and contains at least $\frac{1}{2}(|E(G)| + |M|)$ edges.

   (Hint: Think of a random bipartition scheme of the set of vertices of G such that you guarantee that each edge of the matching has its endpoints on opposite sides of the partition. Calculate the expected number of edges of G that have their endpoints in opposite partitions.)

2. The discrete log problem is : given a prime p and a generator g for the multiplicative group Z_p^*, and a point y chosen at random in Z_p^*, find x such that $g^x = y$. Establish the following claim. Suppose some deterministic poly-time algorithm correctly solves the discrete log problem for a $1/\text{poly}(n)$ fraction of $y \in Z_p^*$ (n is the length of the prime p, i.e. the size of the input); then there is a randomized poly-time algorithm that solves discrete log at all points with high probability.

3. A graph $G = (V, E)$ is called an (n, d, c)-expander if the graph has n vertices with maximum degree d and satisfies the following property: for every subset W of V with $|W| \leq n/2$, W "expands", i.e. the size of the neighbourhood of W (denoted $N(W)$) is large, more precisely $|N(W) \cup W| \geq (1+c)|W|$. Using the probabilistic method, one can show that such graphs exist.

   We can use expanders to amplify the correctness of RP algorithms. Let G be a $(2^n, 5, (2-\sqrt{3})/4)$-expander (there is a known construction for that). Use your n-bits of randomness to pick a random starting point in G. For $\delta = O(\log n)$, find all the nodes $y_1, y_2, \ldots, y_k$ that are within distance δ of your starting node. Run the RP algorithm k times using the y's instead of random strings. Prove that this method will lower the error bound to $1/n^c$ for some constant c.

4. Prove that if $\mathsf{NP} \subseteq \mathsf{BPP}$ then $\mathsf{RP} = \mathsf{NP}$.

5. You go to your local flee market where you find a vender selling dot-product machines. A dot product machine has an unknown $x \in \{0,1\}^n$ kept inside it and an input called $z \in \{0,1\}^n$. On input z, it outputs the GF_2 inner-product of x and z, i.e. it outputs $x_1z_1 + x_2z_2 + \cdots + x_nz_n \mod 2$. The size of the machine is n and there is one for every size. Of course, you want one. But the one you

pick out is too expensive. You beg, you plead, but the vender stands firm. Just as you are about to go, he tells you to wait. From under the counter, he brings out a machine equipped for the input size you need. The price of this particular one is reduced by 90%. "Why is this one so cheap?", you ask. "It's broken." he answers. He goes on to explain that it works fairly well, considering. Obviously a healthy machine M which computes a linear function must always pass the following **linearity test**: $M(z_1) + M(z_2) = M(z_1 + z_2)$. The cheap machine is pretty good, it passes the test 99% of the time over random choices of z_1 and z_2. You can't resist such a deal - you buy it. Now what?

You wish it worked just as well as a healthy machine. However, it is possible to use the cheap, defective machine to simulate a healthy one. Furthermore, for each query to the healthy machine you will only need to make a constant number of queries to the broken machine as part of the simulation.

The basic experiment is as follows. Since you can't just believe M when we query it on z, you calculate $M(z)$ in a roundabout way. We take a *self-correcting sample* at z: Pick a random r, output $M(z+r) + M(r)$. (In the case this is not equal to $M(z)$, the machine is clearly doing something non-linear.)

(a) Show that for any z, when we take a self-correcting sample at z, the probability of getting the same value twice is at least .98. (Hint: Show that $\Pr_{r_1,r_2}[M(z + r_1) + M(r_2) = M(z + r_1 + r_2) = M(r_1) + M(z + r_2)] > .98$.)

(b) Using (a), show that when you take a self-correcting sample at z, you get the same value at least .96 of the time.

(c) Define $M'(z)$ as the value you get at least .96 of the time when you make a self-correcting sample at z. Show that the M' is a function computed by some healthy dot product machine. (Hint: All you need to show is that M' is linear. Fix any z_1 and z_2. Argue that $\Pr_r[(M(r + z_1) + M(r) = M'(z_1)) \cap (M(r + z_2) + M(r) = M'(z_2)) \cap (M(r + z_1) + M(r + z_2) = M'(z_1 + z_2))] > 0$. Conclude therefore it must be true.)

(d) Show that $\Pr_r[M(r) = M'(r)] \geq .94$. Show that M' is the unique linear function which is that close to M. (Hint: Show that $\Pr_{z,r}[M(z) = M(z + r) + M(r) = M'(z)] \geq .94$.)