CS 170 Efficient Algorithms and Intractable Problems
Spring 2017 Prasad Raghavendra and Sanjam Garg

Final Review II-3

1. Cryptography

Let $G = (V,E)$ be a graph that is three-colorable and let c_1, $c_2 : V \to \{R,G,B\}$ be two *valid* coloring functions. c is a valid coloring function if for every $(u,v) \in E$ we have that $c(u) \neq c(v)$. Let $< P,V >$ be the zero-knowledge proof system described in class. Answer True/False for the following questions.

- A malicious verifier learns whether the prover used c_1 or c_2.
- An honest prover in $< P,V >$ can prove to the verifier that the graph is three colorable even without being provided a valid c.
- An honest verifier in $< P,V >$ accepts a proof only if there exists a three coloring of the considered graph.

2. Sampling, Streaming

1. Let $[m]$ denote the set $\{0,1,\cdots,m-1\}$. For each of the following families of hash functions, say whether or not it is universal, and determine how many random bits are needed to choose a function from the family.
   (a) $H = \{h_{a_1,a_2} : a_1,a_2 \in [m]\}$, where m is a fixed prime and $h_{a_1,a_2}(x_1,x_2) = a_1x_1 + a_2x_2 \mod m$. Notice that each of these functions has signature $h_{a_1,a_2:}[m]^2 \to [m]$, that is, it maps a pair of integers in $[m]$ to a single integer in $[m]$.
   (b) H is as before, except that now $m = 2^k$ is some fixed power of 2.
   (c) H is the set of all functions $f : [m] \to [m-1]$.
2. Given a graph $G = (V,E)$ on n vertices, we would like to estimate the fraction of triples $(i,j,k) \in V$ that form a triangle in G, i.e., (i,j), (j,k) and $(k,i) \in E$. Design a randomized algorithm to estimate this fraction within an error of ± 0.01 with probability 0.99. What is the run-time of your algorithm, in terms of n?
3. Given a stream with only one distinct word repeated again and again, can the counting distinct algorithm return a value greater than 1000? If so, what is the probability of this event?

3. Multiplicative Weights

1. In an execution of the weighted majority algorithm with n experts, after t days, the algorithm made ℓ mistakes. What is the largest possible value of the total weight of all experts?
2. In an execution of the weighted majority algorithm with n experts, suppose there is an expert (say E_1) who never makes a mistake. What is the maximum number of mistakes the algorithm will make?
3. "Follow the leader" is a strategy for the experts problem, where on each day, the aggregator always picks the best expert so far, i.e., the expert with the smallest number of mistakes up to day the day.
   Give an example to show that with 3 experts, this strategy could make three times as many mistakes as the best expert.
4. We are running Probabilistic Multiplicative Weights with $n = 2$ experts for $T = 10,000$ days, and we choose $\varepsilon = 0.01$ Is this close to the optimum choice? (Hint: $\sqrt{\ln 2} \approx .8325$)

5. Consider Probabilistic Multiplicative Weights algorithm, in all of the first 140 days, Expert 1 has cost 0 and Expert 2 has cost 1. In the next day, with what probability will you play Expert 1? Assume $\varepsilon = 0.01$. (Hint: You can assume that $0.99^{70} = \frac{1}{2}$)

4. Insider Trading

You are trying to invest in stocks. At the beginning of every day, you buy 1 dollar of stocks, and cash in at the end of the day. We say that a company i has return g_i^t on day t if it earns you a profit of g_i^t dollars per dollar invested.

You have a friend who works at the stock exchange. One day, he hands you a list of n companies (assume n is large), and tells you that for the next T days, one of the companies x (you don't know which one) is going to, on average, earn a profit of 0.5 per day. Additionally, company x will earn a profit of at most 1 per day.

Design a scheme for investing so that at the end of T days, you make at least $0.45T - \frac{\ln n}{0.1}$ dollars. Assume $T \geq 100n$.

Assume for all companies, g_i^t is lower bounded by 0 (you never lose money). Note that the upper bound of 1 dollar only applies to company x.