Probabilistic Algorithms: Exercise 3

Submission Date: December 19th

January 17, 2005

1. (The quality of the LP solution for MIN-SET-COVER) Recall that in the MIN-SET-COVER problem we are given m sets $S_1, \ldots, S_m \subseteq [n]$ that are a cover (that is $\cup S_j = [n]$) and we need to find a minimal sized cover $A \subseteq \{S_1, \ldots, S_m\}$. In class we saw an $O(\log n)$ approximation algorithm. This algorithm used the following linear programming problem.

   $min \sum_{1 \leq j \leq m} y_j$
   $s.t.$
   $\forall i \in [n] : \sum_{j | i \in S_j} y_j \geq 1$

   Let $y_1^*, \ldots, y_m^*$ denote the solution in $\{0,1\}$, and $\hat{y}_1, \ldots, \hat{y}_m$ denote the solution in $[0,1]$. The purpose of this exercise is to exhibit a set-cover problem in which $\sum y_j^* \geq \Omega(\log n) \cdot \sum \hat{y}_j$. (This shows that we can't get a better approximation this way.)

   Consider the following problem. Let t be a number, and consider the complete graph on t vertices (that is every two vertices $1 \leq i, j \leq t$ are connected by an edge). Let n be the number of edges in this graph. For every cut $C \subseteq [t]$ in the graph (recall that a cut is a set C which isn't empty and its complement isn't empty), we define a set $S_C \subseteq [n]$ that is the set of edges in the cut. More formally, $S_C = \{(i,j) : i \in C,\ j \notin C\}$. This defines a set-cover problem.

   (a) Show that $\Sigma \hat{y}_C = O(1)$. That is, that in the optimal real solution the sum does not depend on n.

   (b) Show that any integer solution y^* to this problem defines a one-to-one mapping from $[t]$ to $\{0,1\}^{\sum y_C^*}$. (Hint: for a fixed set S_C consider the mapping from $[t]$ to $\{0,1\}$ defined by whether the input i is in C.)

   (c) Conclude that $\sum y_C^* \geq \Omega(\log n) \sum \hat{y}_C$.

2. (Program checking and correcting for matrix multiplication) In this question all the arithmetic operations are in the field $\mathbb{F}_2$ (the field with two-elements). We are given a (deterministic) program that implements a function $mult$ that takes as input two $n \times n$ matrices A and B. The programmer claims that his program implements very efficient matrix multiplication. Namely that $mult(A,B) = AB$ (the product of A and B over $\mathbb{F}_2$).[1] We don't trust this program. The following questions will provide methods for checking and possibly correcting the program without harming the running time.

   (a) Let $a \in \mathbb{F}_2^n$ be a vector different from zero. Show that $\Pr[< a, x >= 0] \leq 1/2$ where the probability is over choosing a random vector $x \in \mathbb{F}_2^n$ and $< \cdot, \cdot >$ is the inner-product in $\mathbb{F}_2$.

[1] The trivial algorithm takes $O(n^3)$ basic operations, however there are faster (and more complicated) algorithms with time $\approx n^{2.3}$.

(b) Given matrices A, B and C where $C = mult(A, B)$ we want to verify that $AB = C$. Give a simple probabilistic procedure that performs $O(n^2)$ basic operations such that for every A, B, C:

- if $AB = C$ then the procedure accepts with probability 1.
- if $AB \neq C$ then the procedure rejects with probability at least $1/2$.

Note: One simple possible solution could have been to compute AB however, this takes too much time.

(c) Assume that we know that the procedure works well on average A and B. That is that $\Pr[mult(A, B) = AB] \geq 7/8$ where the probability is over choosing random matrices over $\mathbb{F}_2$. Show how to correct the program to a probabilistic algorithm $mult'$ such that for any input A, B, $\Pr[mult'(A, B) = AB] \geq 1/2$ where the probability is over the coin tosses of $mult'$. You are allowed to use $mult$ as a procedure a constant number of times and to use $O(n^2)$ additional basic operations. (Hint1: First show how to obtain an algorithm $mult''$ that works well for any matrix A when B is uniformly selected. That is, $\forall A$, $\Pr_B[mult''(A, B) = AB] \geq 1/2$. Hint2: To implement Hint1, think how to reduce the problem of computing AB to a constant number of other matrix multiplications where the second matrix is uniformly selected.)