COS 341: Discrete Mathematics

Homework #8 — Fall 2006
Chernoff, Walks, Numbers — Due: Friday, December 8

See instructions on the "assignments" web-page on how and when to turn in homework, and be sure to read the collaboration and late policy for this course. Approximate point values are given in brackets. *Be sure to show your work and justify your answers.*

1. In a recent election, there were two candidates, Batman and Penguin; n citizens cast their votes for Batman, and m cast their votes for Penguin, where $n > m$. However, as is so often the case, the voting machines sometimes malfunctioned. In particular, each vote cast for Batman was registered (and thus counted) as a vote for Penguin with probability $p < 1/2$, and each vote cast for Penguin was similarly registered as a vote for Batman with the same probability p. You can assume that whether or not the machine malfunctioned on any particular citizen's vote was (mutually) independent of whether or not it malfunctioned on other citizens' votes.

(Note the important distinction between how a vote was *cast*, meaning how a citizen *intended* for the vote to be counted, and how a vote was *registered*, meaning how the vote was *actually* counted by the voting machine.)

a. [5] Use Chernoff bounds to compute an upper bound on the probability that Batman does not win the election after counting all the votes as they were registered by the voting machines. This probability should be a function of n, m and p. You can use which ever Chernoff bounds you wish, whether from the notes or lecture, provided that the assumptions of the bound you are using are appropriately satisfied. Also, you may wish to look ahead to the next part so that you choose a bound that will make it easier to answer that question.

b. [5] Given a small positive number δ, what is the largest value of p for which your bound in the last part is smaller than δ?

c. [3] The "actual margin of victory" is the fraction of the citizens who cast votes for Batman, minus the fraction who cast votes for Penguin. What is the largest probability p of voting-machine malfunction that can be tolerated in a state-wide election of a million voters when the actual margin of victory is 1%, assuming we want to be 99.9% confident that the true winner (Batman) gets the most registered votes? Answer the same question for a national election with 100 million voters when the actual margin of victory is 0.5%.

2. Alice and Bob are playing the following coin-flipping game. First, Alice selects a three-toss pattern, such as `TTH`. Next, Bob selects a different three-toss pattern, say `HTT`. Finally, they start flipping a fair coin repeatedly until one of the two selected patterns is observed, at which point, the person who chose the observed pattern is declared the winner. For instance, suppose after making the choices above that the coin tosses come up `THHTHTHHTT`. At this point, the game would end, since `HTT` was observed, and Bob, who had selected that pattern, would be the winner.

a. [5] For the choices of patterns given above, what is the probability that Alice wins the game?

b. [5] For the choices of patterns given above, what is the expected number of tosses until the game ends?

3. A boolean variable x can have two possible values, TRUE or FALSE. A 2-SAT formula is an AND of a set of clauses. Each clause is an OR of two literals, where each literal is a boolean variable x_i or its negation $\overline{x}_i$. For example,

$$(x_1 \vee \overline{x}_2) \wedge (\overline{x}_1 \vee \overline{x}_3) \wedge (x_2 \vee x_3)$$

is a 2-SAT formula. A solution to a 2-SAT formula is a $\{\text{TRUE}, \text{FALSE}\}$-assignment to the variables $x_1, x_2, \ldots, x_n$ such that all the clauses are satisfied. For instance, the truth assignment that sets $x_1 = x_2 = \text{TRUE}$ and $x_3 = \text{FALSE}$ is a solution to the formula above. In this problem, we analyze a simple randomized algorithm for solving 2-SAT. Here, n is the number of variables and m is an integer parameter.

Algorithm 2-SAT

a. Start with an arbitrary truth assignment.

b. Repeat up to $2mn^2$ times, terminating if all clauses are satisfied:

   (i) Choose an arbitrary clause that is not satisfied.

   (ii) Choose uniformly at random one of the literals in the clause and switch the value of its variable.

c. If a valid truth assignment has been found, return it.

d. Otherwise, return that the formula is unsatisfiable.

We refer to each change in the truth assignment as a step and analyze the number of steps executed by the algorithm. Henceforth, assume that the formula is satisfiable and let S be a particular satisfying assignment for the formula (pick an arbitrary one if several exist). Let A_i represent the variable assignment after the ith step of the algorithm. Let X_i denote the number of variables in the current assignment A_i that have the same value as in the satisfying assignment S. When $X_i = n$, the algorithm certainly terminates with a satisfying assignment. (In fact, the algorithm could terminate earlier if it finds another satisfying assignment.) Starting with $X_0 < n$, we consider how X_i changes in every step.

a. [5] Prove that if $X_i = j$, then X_{i+1} is either $j+1$ or $j-1$. Further show that

$$\begin{aligned}
\Pr[X_{i+1} = 1 \mid X_i = 0] &= 1 \\
\Pr[X_{i+1} = j+1 \mid X_i = j] &\geq 1/2 \quad \text{for } 1 \leq j \leq n-1 \\
\Pr[X_{i+1} = j-1 \mid X_i = j] &\leq 1/2 \quad \text{for } 1 \leq j \leq n-1
\end{aligned}$$

b. [5] Consider a random walk on $0, 1, \ldots, n$ where the position after step i is Y_i. The initial position $Y_0 = X_0$, and

$$\begin{aligned}
\Pr[Y_{i+1} = 1 \mid Y_i = 0] &= 1 \\
\Pr[Y_{i+1} = j+1 \mid Y_i = j] &= 1/2 \quad \text{for } 1 \leq j \leq n-1 \\
\Pr[Y_{i+1} = j-1 \mid Y_i = j] &= 1/2 \quad \text{for } 1 \leq j \leq n-1.
\end{aligned}$$

The expected time for Y_i to reach n is an upper bound on the expected time for X_i to reach n. (You may assume this). Compute the expected number of steps for Y_i to reach n.

c. [2.5] Show that the probability that the 2-SAT algorithm terminates within $2n^2$ steps is at least $1/2$.

d. [2.5] Show that the probability that the 2-SAT algorithm terminates within $2mn^2$ steps is at least $1 - 1/2^m$.

4. Prove the following assertions:

a. [2.5] For all $c \neq 0$, $a|b$ if and only if $ca|cb$.

b. [2.5] Every common divisor of a and b divides $\gcd(a, b)$.

c. [3] $\gcd(\text{rem}(a, b), b) = \gcd(a, b)$. (Hint: First prove the more general fact that $\gcd(a - qb, b) = \gcd(a, b)$ for all integers q.)

5. SpongeBob and Patrick are playing a game. They start with two distinct, positive integers written on a blackboard — call them a and b. They then take turns, with SpongeBob going first. On each player's turn, he must write a new positive integer on the board that is the difference of two numbers that are already there. If a player can not play, then he loses. For example, suppose that 12 and 15 are on the board initially. SpongeBob's first play must be 3, which is $15 - 12$. Then Patrick might play 9, which is $12 - 3$. Then SpongeBob might play 6, which is $15 - 9$. Then Patrick cannot play, so he loses.

a. [4] Prove that every number on the board at the end of the game is a multiple of $\gcd(a, b)$.

b. [5] Prove that every positive multiple of $\gcd(a, b)$ up to $\max(a, b)$ is on the board at the end of the game.

c. [3] If each player plays as cleverly as possible, who will win?