CS5314 Randomized Algorithms

Assignment 1
Due: 1:10 pm, Oct 26, 2010 (before class)

1 Questions

1. (20%) We draw cards uniformly at random without replacement from a standard deck of cards. Let A be the event that the first two cards are red, and B be the event that the third card is black.

   (a) Compute $\Pr(A)$.

   (b) Compute $\Pr(B)$.

   (c) Compute $\Pr(B \mid A)$.

   (d) Compute $\Pr(A \mid B)$. *Hint:* Use the results of (a), (b), and (c).

2. (10%) There are three special dice, having the following numbers on their faces:

   - die A - 1, 2, 2, 2, 3, 3
   - die B - 1, 1, 2, 2, 3, 3
   - die C - 1, 1, 2, 2, 2, 3

   All three dice are *fair*, so that for each die, each face comes up with probability 1/6.

   (a) You are going to throw die A and die C. What is the probability that the outcome of die A is greater than the outcome of die C?

   (b) You throw the three dice in an unknown order, and you get two 2's and one 3. What is the probability that the die with outcome 3 is die C?

3. (20%) Let X and Y be independent geometric random variables, where X has parameter p and Y has parameter q.

   (a) What is the probability that $X = Y$?

   (b) What is $\mathrm{E}[\max(X, Y)]$?

   (c) What is $\Pr(\min(X, Y) = k)$?

   (d) What is $\mathrm{E}[X \mid X \leq Y]$?

4. (10%) We draw cards uniformly at random with replacement from a deck of n cards. Suppose we draw the cards $2n$ times.

   (a) What is the expected number of cards that are not chosen at all?

   (b) What is the expected number of cards that are chosen exactly once?

5. (10%) Recall the branching process problem described in Lecture Notes 5. Suppose that each program P, before it finishes running, will create 2 new copies of itself with probability p, and 0 new copies with probability $1 - p$.

(a) Let Y_i denote the number of copies of P in the ith generation, where $Y_0 = 1$. Determine $\mathrm{E}[Y_i]$.

(b) For what values of p will the expected total number of copies be bounded?

6. (30%) You need a new staff assistant, and you have n people to interview. You want to hire the best candidate for this position. When you interview the candidates, you can give each of them a distinct score, so that the one with the highest score will be the best.

   You interview the candidates one by one. Because of your company's hiring policy, after you interview the kth candidate, you either offer the candidate the job immediately, or you will forever lose the chance to hire that candidate.

   We suppose that the candidates are interviewed in a random order, chosen uniformly at random from all $n!$ possible orderings.

   Consider the following strategy: First interview m candidates but reject them all. Then, from the $(m+1)$th candidate, hire the first candidate who is better than *all* of the previous candidates you have interviewed.[1]

   (a) Let E be the event that we hire the best candidate, and let E_i be the event that the ith candidate is the best and we hire him. Determine $\Pr(E_i)$, and show that

   $$\Pr(E) = \frac{m}{n} \sum_{j=m+1}^{n} \frac{1}{j-1}.$$

   (b) Show that

   $$\Pr(E) \geq \frac{m}{n}(\log_e n - \log_e m).$$

   (c) Show that

   $$\frac{m}{n}(\log_e n - \log_e m) \text{ is maximized}$$

   when $m = n/e$, and conclude that $\Pr(E) \geq 1/e$ for this choice of m.

2 Further Studies (No marks)

1. Prove that if $E_1, E_2, \ldots, E_n$ are mutually independent, then the events $\overline{E_1}, \overline{E_2}, \ldots, \overline{E_n}$ are also mutually independent.

2. During the class, we have studied a simple randomized algorithm so that given any graph, we can find its min-cut with probability at least $2/(n(n-1))$. Now, we define an r-cut of a graph G to be a set of edges in G whose removal will break G into r or more connected components. (That is, the normal definition of a cut is equivalent to a 2-cut here.) Describe a randomized algorithm for finding an r-cut with minimum number of edges. Also, analyze the probability that the algorithm succeeds in one iteration.

[1] That is, you will hire the kth candidate if $k > m$ and this candidate is better than all of the $k-1$ candidates you have already interviewed.