CS174 Midterm 1 Solutions

1. Suppose that we have a biased six-sided die, and we wish to use it to generate uniform binary random numbers. Consider the following algorithm:

   - Throw the die twice.
   - If the first throw is a 1 and the second is a 2, output 0.
   - If the first throw is a 2 and the second is a 1, output 1.

   Suppose that the probability distribution of the die outcome X satisfies $\Pr(X = i) = p_i$ with $0 < p_i < 1$ for $i = 1, \ldots, 6$.

   (a) Verify that, if the algorithm outputs a number, it is chosen uniformly from $\{0, 1\}$.
   Suppose we repeatedly call the algorithm until it outputs a number. What is the expected number of die throws required? What is the variance of the number of die throws required?

   Let X_1, X_2 be the outcomes of the two throws. The probability that the algorithm outputs a 0 is p_1p_2. The probability that it outputs a 1 is p_2p_1. Thus, the conditional probability that it outputs a 0 given that it outputs a number is

   $$p_1p_2/(2p_1p_2) = 1/2.$$

   The probability that the algorithm outputs a number is $2p_1p_2$. Thus, the number of calls required has a geometric distribution with parameter $2p_1p_2$. This has expectation $1/(2p_1p_2)$. The variance of this geometric random variable is $(1 - 2p_1p_2)/(4p_1^2p_2^2)$. Since each call requires two throws, the expected number of throws is $1/(p_1p_2)$, and the variance of the number of throws is $2^2 \times (1 - 2p_1p_2)/(4p_1^2p_2^2) = (1 - 2p_1p_2)/(p_1^2p_2^2)$.

(b) Can you suggest an alternative algorithm that uses fewer die throws to generate a uniform bit? That is, propose an alternative algorithm, show that it generates uniform bits, and show that the expected number of throws per random bit is smaller than the previous algorithm. Is the variance of the number of throws per random bit always strictly smaller than the previous algorithm?

Throw the die twice and then if the first outcome is in $\{1,3,5\}$ and the second in $\{2,4,6\}$, output 0, otherwise if the first outcome is in $\{2,4,6\}$ and the second in $\{1,3,5\}$, output 1. The proof that the distribution is uniform is as before. The number of throws is again a geometric random variable, but this time the parameter is $2p_{\text{odd}}p_{\text{even}}$, where $p_{\text{odd}} = p_1+p_3+p_5$ and $p_{\text{even}} = p_2+p_4+p_6$. The expected number of throws is $1/(2p_{\text{odd}}p_{\text{even}})$, which is a decreasing function of the parameter, and so is clearly larger than $1/(2p_1p_2)$, since $p_3+p_5 > 0$ and $p_4+p_6 > 0$.

In addition, the variance is $1-2p_{\text{odd}}p_{\text{even}}/(4p_{\text{odd}}^2p_{\text{even}}^2)$. Again the variance is a decreasing function of the parameter, so in this case it is smaller than in the previous case.

(c) Suppose that we wanted to use the die to generate random numbers distributed uniformly on $\{0, 1, 2\}$. Suggest a suitable algorithm.

There are many possibilities. For example, we could use the earlier algorithm for random bits, and then if we get the binary representation for $0, 1, 2$, we output that, otherwise we try again. Alternatively, we could throw the die three times, and output 0 if the sequence is 1 then 2 then 3, 1 if it is 2 then 3 then 1, and 2 if it is 3 then 1 then 2.

2. Let $X_1, X_2, \ldots$ be a sequence of independent identically distributed random variables taking values in $[0,1]$. Consider the following online, randomized algorithm for estimating the expectation $\mathbb{E}[X_1] = \mathbb{E}[X_t]$. At time t, the algorithm is called with inputs (t, X_t, Y_{t-1}) and it outputs a single bit, $Y_t \in \{0,1\}$ (and we define $Y_0 = 0$). To choose its output, the algorithm tosses a biased coin: with probability $(1-1/t)$, it outputs Y_{t-1}, otherwise it tosses a second biased coin, and, with probability X_t, it returns 1, otherwise it returns 0.

   (a) Show that the output produced by the algorithm at each step is an unbiased estimate of the expectation of the X_i, that is,
$$\mathbb{E}[Y_t] = \mathbb{E}[X_1].$$

   Let A_t be the first biased coin tossed by the algorithm in round t, and B_t the second. Clearly,
$$\begin{aligned}\mathbb{E}[Y_t] &= \mathbb{E}[Y_t|A_t = 1]\Pr(A_t = 1) + \mathbb{E}[Y_t|A_t = 0]\Pr(A_t = 0)\\ &= \mathbb{E}[Y_{t-1}]\left(1 - \frac{1}{t}\right) + \frac{\mathbb{E}[X_t]}{t}.\end{aligned}$$

   Now, since $Y_0 = 0$, $\mathbb{E}[Y_1] = \mathbb{E}[X_1]$. If $\mathbb{E}[Y_{t-1}] = \mathbb{E}[X_1]$, then
$$\begin{aligned}\mathbb{E}[Y_t] &= \mathbb{E}[X_1]\left(1 - \frac{1}{t}\right) + \frac{\mathbb{E}[X_t]}{t}\\ &= \mathbb{E}[X_1].\end{aligned}$$

   The result follows by induction.

(b) What is the variance of Y_t?

Since Y_t is a Bernoulli random variable with parameter $p = \mathbb{E}[Y_t] = \mathbb{E}[X_1]$, its variance is

$$p(1-p) = \mathbb{E}[X_1]\left(1 - \mathbb{E}[X_1]\right).$$

3. Consider a complete directed graph G with $n+1$ vertices, labelled $\{0, \ldots, n\}$. Suppose that we have a set of n colors and, for each vertex v in G, the n edges leading out of v each are colored with a distinct one of the n colors. Call such a graph G a *complete n-colored map.* Notice that we can use a finite sequence of colors $(c_1, \ldots, c_T)$ to travel around G: we start at vertex $v_0 = 0$ at time 0 and, at time t, move from vertex v_{t-1} to vertex v_t by traversing the edge with color c_t. A *search sequence* for G is a color sequence for which the sequence of visited vertices includes every vertex in G. A *universal search sequence* is a color sequence that is a search sequence for every complete n-colored map.

   Consider a random color sequence $(c_1, c_2, \ldots, c_T)$, in which each c_t is chosen independently and uniformly from $\{1, \ldots, n\}$.

   (a) Give an upper bound on the probability that this sequence is not a universal search sequence.

   First, fix a complete n-colored map G. The probability that the sequence is not a search sequence for G is the probability that some node is never visited. For a given node, the probability that it is never visited is $(1 - 1/n)^T$. By the union bound (Bonferroni inequality), the probability that the sequence is not a search sequence is no more than $n(1-1/n)^T$. Again by the union bound, the probability that the sequence is not a universal search sequence is no more than $Nn(1 - 1/n)^T$, where N is the number of distinct complete n-colored maps. But clearly $N = (n!)^{n+1}$. Thus,

$$\Pr((c_1, \ldots, c_T) \text{ not a univ. search seq.}) \leq (n!)^{n+1} n(1 - 1/n)^T.$$

(b) Show that if $T = \Omega(n^3 \log n + n \log(1/\delta))$, the probability that the sequence is a universal search sequence is at least $1 - \delta$.

Taking logs, and using the fact that $\log x \leq x - 1$ for $x > 0$, we have that the probability that the sequence is not a universal search sequence is no more than δ for

$$\begin{aligned} (n+1)\log(n!) + \log n + T\log(1-1/n) &\leq \log(\delta) \\ \Leftarrow \qquad (n+1)^2 \log(n) - T/n &\leq \log(\delta) \\ \Leftarrow \qquad T \geq (n+1)^3 \log(n) + n\log(1/\delta). \end{aligned}$$