Assignment 5

15-359 Probability and Computing

Name: Shashank Singh
Email: sss1@andrew.cmu.edu
Section: B
Due: Friday, February 24, 2012

Problem 1: Not too concentrated

Let $f_X : \mathbb{R} \to \mathbb{R}$ be the probability density function of X, so that $f_X \leq B$. Then, $\forall x, \epsilon \in \mathbb{R}$ with $\epsilon > 0$,

$$\begin{aligned} P(|X - x| \leq \epsilon) &= P(x - \epsilon \leq X \leq x + \epsilon) \\ &= \int_{x-\epsilon}^{x+\epsilon} f_X(t)\, dt \\ &\leq \int_{x-\epsilon}^{x+\epsilon} B\, dt \\ &= B(x + \epsilon) - B(x - \epsilon) = \boxed{2\epsilon B.} \end{aligned}$$

Problem 2: Mediocristan and Extremistan

A. Let $\lambda = 0.1, \alpha = \frac{10}{9}$. Then, $M \sim \text{Exp}(\lambda)$ (M has an exponential distribution), and $X \sim \text{Pareto}(\alpha)$ (X has a Pareto distribution). As shown in class, $E[M] = \boxed{\frac{1}{\lambda}.}$ By definition of expected value, since $\alpha > 1$,

$$\begin{aligned} E[X] &= \int_1^\infty x \cdot f_X(x)\, dx \\ &= \int_1^\infty \alpha x^{-\alpha}\, dx \\ &= -\alpha \left(\frac{1^{1-\alpha}}{1 - \alpha} \right) = \boxed{\frac{\alpha}{\alpha - 1}.} \end{aligned}$$

B. Since, $\forall m \geq 0$, $\overline{F}_M(m) = P(M > m)$, and, $\forall x \geq 1$, $\overline{F}_X(x) = P(X > x)$,

$$\frac{1}{100} = e^{-\lambda m_0},$$

and

$$\frac{1}{100} = x_0^{-\alpha}.$$

Thus, $\boxed{m_0 = 10 \ln(100) \approx 46,}$ and $\boxed{x_0 = 10^{9/5} \approx 63.}$

C.
$$\rho_M = \lambda^2 \int_{10 \ln 100}^{\infty} m f_M(m)\, dm \approx 5.6\%.$$
$$\rho_X = \frac{\alpha - 1}{\alpha} \int_{10^{9/5}}^{\infty} x f_X(x)\, dm \approx 63\%.$$

Thus, $\rho_M < \rho_X$. This makes sense because the exponential distribution decays much more quickly than the Pareto distribution. ■

Problem 5: Vanilla search trees

A. Randomly permute the elements of A. Insert the elements into a binary search tree as usual.

B. A clever proof is to notice that the problem is isomorphic to analyzing the expected depth of an element the recursion tree of randomized quicksort; the depth of each node in the tree corresponds to the level of recursion at which it is used as a pivot in the algorithm. The expected depth of this tree was shown in class to be at most $2H_n$, where H_n denotes the n^{th} harmonic number.

A more rigorous, albeit boring, proof is as follows: Let X be a random variable denoting the average expected distance of a node in the tree from the root. $\forall i \in \{1, 2, \ldots, n\}$, let X_i be a random variable denoting the distance of a_i from the root. $\forall (i, j) \in \{1, 2, \ldots, n\}^2$ with $i \neq j$, let $X_{i,j}$ be an indicator random variable which is 1 if ${}_j j$ is an ancestor of a_i in the tree. Note that, since the depth of a node is the same as the number of ancestors it has, $\forall i \in \{1, 2, \ldots, n\}$, $X_i = \sum_{j=1}^{n} X_{i,j}$. The a_j is a parent of a_i if and only if a_j is the first elements of $\{a_j, a_{j+1}, \ldots, a_i\}$ to be inserted into the tree. This happens with probability $\frac{1}{|j-i|+1}$, so that,

$$X = \frac{1}{n} \sum_{i=1}^{n} X_i = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{1}{|j-i|+1} \leq \frac{1}{n} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \frac{2}{j-i+1}.$$

This summation can be re-written in terms of $k = j - i$ as

$$E[X] = \frac{1}{n} \sum_{k=2}^{n} \sum_{l=0}^{n-k+1} \frac{1}{k} = \frac{1}{n} \sum_{k-2}^{n} (n-k+1) \frac{2}{k} \leq \sum_{k-2}^{n} \frac{2}{k} \leq 2H_n \in O(\log n),$$

where H_n denotes the n^{th} harmonic number. ■

Problem 6: Have randomness, will predict

Let $f : \mathbb{N} \to [0, 1] \subseteq \mathbb{R}$ such that, $\forall n \in \mathbb{N}$, $f(x) = 1 - e^{-x}$. Let T be a real number randomly sampled from $[0, 1]$ (in particular, let $T \sim U(0, 1)$). If $T > f(B)$, then guess that $B < Z$. Otherwise, guess that $B > Z$.

Let $W = \min\{X, Y\}$, let $V = \max\{X, Y\}$, let C be the probability that you guess correctly. Since f is strictly increasing on its domain, $f(W) < f(V)$, so that

$$\begin{aligned} P(C) &= P(B = W)P(f(B) < T) + P(B = V)P(T < f(B)) \\ &= \frac{1}{2}(1 - f(W)) + \frac{1}{2}f(V) \\ &= \frac{1}{2} + \frac{1}{2}\left(f(V) - f(W)\right) > \frac{1}{2}. \end{aligned}$$

Thus, guessing in this manner guarantees that you will guess correctly with probability $\frac{1}{2} + \alpha$, for some $\alpha > 0$. ■

Problem 7: Shooting blanks

A. A deterministic algorithm $A \in \mathcal{A}$ consists of querying the entries of an input matrix in some order, returning `true` upon finding that any column contains only 0's, and returning `false` after confirming that no column in A contains only 0's.

B. No optimal algorithm will query an entry in the matrix more than once or query any entry in a column in which a 1 has already been found. Thus, we consider only those algorithms which do not do this (we can do this because we are interested only in those algorithms which minimize $\min_{A \in \mathcal{A}} E[T_A(I_\tau)]$.

   Let I be a boolean matrix with exactly 1 non-zero entry in each column, and let $A \in \mathcal{A}$. Let X be a random variable denoting the number of entries in I queried by A when run on I, and, $\forall i \in \{1, 2, \ldots, n\}$, let X_i be a random variable denoting the number of entries A queries in the i^{th} column. Then, $\forall i, j \in \{1, 2, \ldots, n\}$, $P(X_i = j) = \frac{j}{n}$.
   $E[X_i] = \sum_{j=1}^{n} j \cdot P(X_i = j) = \frac{1}{n}\sum_{i=1}^{n} j = \frac{n+1}{2}$
   Thus, by Linearity of Expectation, since $X = \sum_{i=1}^{n} X_i$,

$$E[X] = E\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} E[X_i] = \frac{n(n+1}{2}.$$

   Therefore, the expected number of queries made by A is at least $\frac{n(n+1)}{2}$.

   Suppose L is a Las Vegas algorithm. Then, by Yao's Minimax Principle the expected number of queries made by L in the worst case is bounded below by $\frac{n(n+1)}{2}$. ■