CS5330: Optional problems for Weeks 1 through 5

February 16, 2020

I strongly encourage you to think about these problems and to discuss with me, your TA, or your peers if you need help. I will post solution hints in about 1 week.

1. Suppose you repeatedly choose an integer from $\{1, \ldots, n\}$ uniformly at random and independently. Give an asymptotic bound (in Θ notation) on the expected number of draws before you choose a number you have already chosen before.

2. A *dominating set* in a graph $G = (V, E)$ is a set of vertices D such that each of the n vertices in V is either in D or adjacent to a vertex in D.

   Suppose we have a d-regular graph in which every vertex has exactly d neighbors. Let D_1 be a random subset of V in which each vertex appears independently with probability p. Let D be the union of D_1 and the set of all vertices that are not adjacent to any vertex in D_1. Clearly, D is a dominating set.

   What value of p minimizes $\mathbb{E}[D]$? Using that value of p, show that the graph must have a dominating set of size $O((n \log d)/d)$.

3. A *common subsequence* of two sequences x and y is a sequence z such that there exist indices $i_1 < i_2 < \cdots < i_k$ and $j_1 < j_2 < \cdots < j_k$ with $x_{i_1} = y_{j_1}, x_{i_2} = y_{j_2}, \ldots, x_{i_k} = y_{j_k}$. For example, *ardab* is a common subsequence of *abracadabra* and *cardtable*.

   Let x and y be words of length n over an alphabet of size n drawn independently and uniformly at random. Give the best upper bound you can on the expected length of the longest common subsequence of x and y. Use the inequality $\binom{n}{k} \le (en/k)^k$.

4. Consider the following algorithm for computing the maximum of an array A. First, A is randomly permuted. Then, you scan from left to right, keeping track of the current maximum. More precisely, a variable m is initialized to $-\infty$, and then for $i = 1, \ldots, n$, if $A[i] > m$, m is set to $A[i]$. Finally, m is returned.

What is the expected number of times m is updated by this algorithm?

5. Consider the random graph $G_{n,p}$ when $p = cn^{-2/3}$, and let X be the number of 4-cliques in the graph.

   (a) What is $\mathbf{E}[X]$?

   (b) What is an upper bound on $\mathsf{Var}[X]$?

6. Suppose you can get iid samples of a random variables X. We want to estimate $\mu = \mathbb{E}[X]$. Suppose you have the information that $\sqrt{\mathsf{Var}[X]}/\mathbb{E}[X]$ is bounded by r. Design an algorithm that uses $O(r^2\epsilon^{-2}\log(1/\delta))$ samples of X to get an estimate $\hat{\mu}$ such that $(1-\epsilon)\mu \le \hat{\mu} \le (1+\epsilon)\mu$.

7. In the *hitting set* problem, the input consists of sets $S_1, \dots, S_n$ that are subsets of $[m] := \{1, \dots, m\}$. The goal is to find a set $T \subseteq [m]$ such that T has nonzero intersection with each S_j. The following integer programming formulation is equivalent. Over integer-valued variables $t_1, \dots, t_m$, minimize $\sum_{i=1}^m t_i$ such that: (i) $\sum_{i \in S_j} t_i \ge 1$ for each $j \in [n]$ and (ii) $t_i \in \{0,1\}$ for each $i \in [m]$.

   Solving integer programs is NP-hard. So, we consider the following linear programming formulation: over real variables $t_1, \dots, t_m$, minimize $\sum_{i=1}^m t_i$ such that: (i) $\sum_{i \in S_j} t_i \ge 1$ for each $j \in [n]$ and (ii) $0 \le t_i \le 1$ for each $i \in [m]$.

   (a) Suppose you solve the above linear program (LP) to get a fractional solution t^*, and let $\ell = \sum_i t_i^*$. Use randomized rounding to obtain a set T, meaning you put each $i \in [m]$ into T with probability t_i independently. What is $\mathbb{E}[|T|]$?

   (b) Argue that any S_j does not intersect with T with probability at most $1/e$. (Hint: Use constraint (i).)

   (c) Design a randomized algorithm that with probability $2/3$, returns a set $\hat{T}$ such that $\hat{T}$ is a hitting set and $|\hat{T}| \le O(\log n) \cdot |T^*|$ where T^* is an optimal solution (i.e, hitting set of minimal size).

8. You are given an array A containing n numbers in sorted order. In one step, an algorithm specifies an integer $i \in [n]$ and is given the value of $A[i]$. Show lower and upper bounds on the expected number of steps taken by a Las Vegas algorithm to determine whether or not a given key k is present in the array.