15-850: Advanced Algorithms **CMU, Spring 2023**
Exercises with Homework #0

Exercises are for fun and edification, please do not submit. (You may discuss the *exercises* with others.) The ones below are grouped by topic and their subparts do not necessarily build on one another (for example, you do not need to do (a) to do (b)).

1. (a) Given random variables (r.v.s) X, Y, show that $\mathbf{E}[X+Y] = \mathbf{E}[X] + \mathbf{E}[Y]$ and $\mathbf{E}[cX] = c\,\mathbf{E}[X]$ and $\mathbf{Var}(cX) = c^2\,\mathbf{Var}(X)$ for any constant c. If they are independent, then show that $\mathbf{E}[XY] = \mathbf{E}[X]\,\mathbf{E}[Y]$ and $\mathbf{Var}(X+Y) = \mathbf{Var}(X) + \mathbf{Var}(Y)$. Hence show that for independent $X_1, \ldots, X_n$, if each X_i has mean μ and variance σ^2, then $\frac{\sum_{i=1}^n X_i}{n}$ has mean μ and variance σ^2/n.

   (b) Let $x_1, x_2, \ldots, x_n$ be a random permutation of the numbers $[n] := \{1, 2, \ldots, n\}$. You scan the numbers from left to right. You have a buffer B of size b where $1 \leq b \leq n$, initially containing b copies of ∞. When you see x_i, if B contains a number bigger than x_i, then drop the largest number in B, and insert x_i to B. What is the expected total number of insertions to B during the scan?

   (c) An airplane in Politesville has n seats, and n passengers assigned to these seats. The first passenger to board gets confused, and sits down at a uniformly random seat. The rest of the passengers do the following: when they board, if their assigned seat is free they sit in it, else (being too polite) they choose a uniformly random empty seat and sit in it. (a) Show that the last person to board sits in their assigned seat with probability 1/2. (b) Show that the expected number of people who board to find their assigned seat occupied is H_{n-1}.

   (d) You want to sample uniformly at random from the set of all n-bit strings that are *balanced*, i.e., that contain exactly $n/2$-many 0's and $n/2$-many 1's (assume n is even). You do the following: take a uniform random sample ω from the set of all n-bit strings. output ω if it is balanced, else reject, and sample again. Show that the expected number of times you sample an ω before outputting a balanced string is $O(\sqrt{n})$. (Hint: Stirling's approximation. If that seems tricky, you can first try to show the expected number of samples is $\leq n+1$—you don't need any fancy approximations for this, only reasoning from first principles.)

2. (a) Given a tree $T = (V, E)$ and a set of nodes $X \subseteq V$ that contains all the leaves, prove that the average degree of nodes in X is less than 2.

   (b) For a graph G, consider two edge-weight functions w_1 and w_2 such that

   $$w_1(e) \leq w_1(e') \qquad \iff \qquad w_2(e) \leq w_2(e')$$

   for all edges $e, e' \in E$. Show that T is an MST wrt w_1 iff it is an MST wrt w_2. (In other words, only the sorted order of the edges matters for the MST.)

   (c) Suppose graph G has integer weights in the range $\{1, \ldots, W\}$, where $W \geq 2$. Let G_i be the edges of weight at most i, and κ_i be the number of components in G_i. Then show that the MST in G has weight exactly $n - W + \sum_{i=1}^{W-1} \kappa_i$.

   (d) Show that if the edge weights are all distinct, there is a unique MST.

3. You are give a bipartite graph $G = (U, V, E)$ with $|U| = |V| = n$ and maximum degree Δ. Parts (b) and (c) are slightly more challenging.

(a) Give an algorithm to color the edges of G with 2Δ colors so that the edges incident to every vertex have distinct colors.

(b) Give an algorithm to color the edges of G with Δ colors so that the edges incident to every vertex have distinct colors. (Hint: matchings.)

(c) Let Δ' be the smallest power of 2 such that $\Delta' \geq \Delta$. Given an $O(n\Delta \log \Delta)$-time algorithm to color the edges with Δ' colors. (Hint: can you create two problems with half the degree at each step?)

4. Given an $m \times n$ matrix A, let A_i denote its i^{th} row and A^j denote its j^{th} column; as always A_{ij} denotes the $(i,j)^{th}$ entry.

   (a) For $m \times n$ and $n \times p$ matrices A and B respectively, their product is an $m \times p$ matrix $C = AB$ where $C_{ij} = \sum_{k=1}^{n} A_{ik}B_{kj} = A_i B^j$. Show that

   $$C = \sum_{k=1}^{n} (A^k B_k).$$

   Observe that each term on the right is the product of a $m \times 1$ column vector with a $1 \times p$ row vector to produce an $m \times p$ matrix.

   (b) There are at least two ways that one can represent a subspace W in $\mathbb{R}^d$. The first is by a set of generators: We say that the vectors $P_1, \ldots, P_k \in \mathbb{R}^d$ are *generators* for the subspace if $W = \{\alpha_1 P_1 + \cdots + \alpha_k P_k \mid \alpha_i \in \mathbb{R}\}$. A second way to represent the subspace W is by a set of constraints: Let $A \in \mathbb{R}^{n \times d}$ be matrix. We say that A is a *constraint matrix* for the space W if

   $$W = \{x \in \mathbb{R}^d \mid Ax = 0\}$$

   i. Let W be a subspace of $\mathbb{R}^d$ given by generators $P_1, \ldots, P_k \in \mathbb{R}^d$. Explain how to write W via a constraint matrix A.

   ii. Let W be a subspace of $\mathbb{R}^d$ given by a constraint matrix A. Explain how to write W via a set of generators.

5. The *column span* of an $n \times d$ matrix A, $n > d$, is the set of vectors $y \in \mathbb{R}^n$ for which $y = A \cdot x$, for some $x \in \mathbb{R}^d$. A vector is a **positive vector** if all its coordinates are non-negative and at least one is *strictly positive*. Given matrix $A \in \mathbb{R}^{n \times d}$, write a poly-sized linear program whose solution is some positive vector in A's column span, and which is infeasible if there is no such vector.