This HW is going out a week before classes start so that you can get a feel for the course asap, and prepare accordingly. It’s a short HW, with a short deadline (end of week #1 of classes), so that we can get to solving problems on the real material for the course soon.

These problems are solvable using ideas we cover in the first 15 lectures of our undergraduate Algorithms course, plus basic probability and linear algebra courses. You can find links to resources for these topics on the course webpage. Unless specified otherwise, all algorithms should run in poly-time. We’re not asking you to optimize your runtimes, but in general please do so when possible (and reasonable); as algorithm designers, it’s a good habit to strive for optimality.

Please solve the (non-exercise) problems without collaboration. You may discuss the exercises with others. Submissions will be via gradescope, and the link will appear on the course webpage and on Piazza. Also, changes, corrections, and clarifications will also appear on Piazza, so please check it regularly. Changes and corrections are marked in red below.

Exercises

Exercises are for fun and edification, please do not submit. The real problems begin on page 3.

The exercises below are grouped by topic and their subparts do not necessarily build on one another (for example, you won’t need to do (a) to do (b)). Feel free to attempt these questions in any order.

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

(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 \(\infty\). 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) What is the probability that the last person to board sits in their assigned seat? (b) What is the expected number of people who board to find their assigned seat occupied?

(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 \(\omega\) from the set of all \(n\)-bit strings. Output \(\omega\) if it is balanced, else reject, and sample again. Show that the expected number of times you sample an \(\omega\) before outputting a balanced string is \(O(\sqrt{n})\). (Hint: Stirling’s formula.) You should also try to show the expected number of samples is \(\leq n + 1\), by arguments 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 at most 2.

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

\[
\forall e,e' \in E \quad w_1(e) \leq w_1(e') \iff w_2(e) \leq w_2(e')
\]

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 \( \kappa_i \) be the number of components in \( G_i \). Then show that the MST in \( G \) has weight exactly \( n - W + \sum_{i=1}^{W-1} \kappa_i \).

3. You are given a bipartite graph \( G = (U,V,E) \) with \( |U| = |V| = n \) and maximum degree \( \Delta \).

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

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

(c) Let \( \Delta' \) 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 \( \Delta' \) 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 \( C_{ij} = \sum_{k=1}^{n} A_{ik}B_{kj} = A_iB^j \). Show that

\[
C = \sum_{k=1}^{n} (A^kB_k).
\]

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.
Problems

Please solve #1–#3, and either #4 or #5.

1. **(The Centers of Attraction.)** For a path \( P = (V, E) \) with positive edge lengths, define \( d_P(u, v) \) to be the length of the subpath between \( u \) and \( v \) in \( P \) according to these edge-lengths. For a set \( C \), define 
\[
d_P(v, C) := \min_{c \in C} d_P(v, c)
\]
to be the distance of \( v \) to its closest center in \( C \). Given a path \( P = (V, E) \) with \( n \) nodes, and an integer \( k \in \mathbb{Z}_{\geq 0} \), you want to pick a set \( C \) of \( k \) “centers” from \( P \) such that 
\[
\sum_{v \in V} d_P(v, C)
\]
is minimized. Give an algorithm that runs in time \( \text{poly}(k, n) \).

2. **(How Many Elements...)** Let \( U \) be a universe of elements, with \( |U| = n \). Consider a sequence \( S = \{a_1, a_2, \ldots, a_m\} \) of \( m \) items, with each \( a_i \in U \). These may not be all distinct, so suppose there are \( D \leq \min(m, n) \) distinct elements in \( S \). We’d like to create an algorithm which can estimate \( D \) (approximately), without storing all of \( S \). For simplicity, assume \( D \) is a power of 2, and \( D \geq 128 \). Define \( L = \log_2 m \).

Our algorithm uses a set of sub-universes \( \{U^0, U^1, \ldots, U^L\} \). Define \( U^0 = U \), and for each \( i \in \{1, \ldots, L\} \), let \( U^i \) be obtained by independently picking each item from \( U^{i-1} \) with probability \( 1/2 \). Observe that \( U^i \subseteq U^{i-1} \). We will also define a set of subsequences \( \{S^0, S^1, \ldots, S^L\} \). Let \( S^i \) be the subsequence of \( S \) that retains only the elements in \( U^i \). Note that \( S^0 = S \).

Our algorithm considers the number of distinct elements in each of the subsequences \( S^i \). Let \( X^i \) be a random variable (r.v.) denoting the number of distinct elements in subsequence \( S^i \).

(a) First, we will prove that \( X^i \) is likely to take a value near its mean. Define the event 
\[
\mathcal{E}_i := \{|X^i - \mathbb{E}(X^i)| < \frac{\mathbb{E}(X^i)}{4}\}.
\]
Show that 
\[
\mathbb{P}[\mathcal{E}_i] \geq 1 - \frac{16}{\mathbb{E}(X^i)}.
\]

(b) Because \( D \) is a power of 2 and \( D \geq 128 \), there must exist some \( i^* \) such that \( \mathbb{E}(X^{i^*}) = 128 \). Define the event \( \mathcal{F} \) to be \( \mathcal{E}_{i^*-1} \cap \mathcal{E}_{i^*} \cap \mathcal{E}_{i^*+1} \). Show that \( \mathbb{P}[\mathcal{F}] \geq 1/2 \).

(c) Now, we are ready to define our estimation algorithm: *Find any level \( i \) for which \( X^i \in [96, 160] \), and output the estimate \( 2^i \cdot X^i \). (If there are no such levels, output zero.*)

Show that this estimate lies in \([\frac{3}{4}D, \frac{5}{4}D]\) with probability at least \( 1/2 \).

*(A variant of this algorithm can give an estimate in the range \([(1 - \epsilon)D, (1 + \epsilon)D]\), instead of in the range \([(1 - 1/4)D, (1 + 1/4)D]\).*

3. **(Social Distancing.)** A convex polygon \( P \subseteq \mathbb{R}^2 \) is the intersection of half-spaces \( P = \{y \in \mathbb{R}^2 \mid a_i^T y \leq b_i\} \).

(a) Given a point \( x \in \mathbb{R}^2 \), write an expression for its distance from a line \( a_i^T y = b_i \).

(b) Write a linear program to find a circle of largest radius that is contained within the polygon. (Think about what the variables should be? What are the constraints?)
4. (A Matter of Degree.) You are given a directed graph $G = (V,E)$, where $V = [n]$, and two vectors $A, B \in \mathbb{Z}_{\geq 0}^n$, and you want to find a subgraph $H = (V, E')$ of $G$ such that vertex $i$ has $A_i$ edges entering it, and $B_i$ edges leaving it.

(a) Give an algorithm that finds such a subgraph in polynomial time (or reports that such a graph does not exist).

(b) Now the requirements change, and you need to find a strongly connected subgraph $H$ of $G$ with the same constraints. Show that the problem is NP-hard, by reduction from a problem in the collection \{Clique, 3-Coloring, Hamilton Cycle\}. \textit{(A strongly-connected digraph is one where there exists an $x$-$y$ path for each pair of vertices $x,y$.)}

5. (The Utility Company) If you want a bit more of a challenge than the previous problem, you can solve this one instead of \#4. You are given a universe $U$ of $n$ elements, and a collection of subsets $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$, with each $S_i \subseteq U$. A subset $S \in \mathcal{S}$ is covered by set $V \subseteq U$ if $S \subseteq V$. The utility of a set $V \subseteq U$ is defined as

$$\text{util}(V) := \frac{\text{number of sets in } \mathcal{S} \text{ covered by } V}{|V|}.$$ 

(a) Use an $s$-$t$ min-cut algorithm to find a set $V$ with largest utility. (Hint: Can you solve the problem when you know the value $\lambda^*$ of the optimal utility? Then how would you remove this assumption?)

(b) Now consider a variant where given $(U, \mathcal{S})$ and two values $(k, \ell)$, the goal is to find a set $V$ of size exactly $k$, that covers at least $\ell$ subsets. Show that this problem is NP-hard, by reduction from a problem in the collection \{Set Cover, Clique, 3-Coloring\}. 
