601.438 / 601.638 Algorithmic Foundations of Differential Privacy Spring 2025
Homework #1 **Due:** February 20, 2025, 9:00am

Reminder: you may work in groups of up to three people, but must write up solutions entirely on your own. Collaboration is limited to discussing the problems – you may not look at, compare, reuse, etc. any text from anyone else in the class. Please include your list of collaborators on the first page of your submission. Many of these problems have solutions which can be found on the internet – please don't look. You can of course use the internet (including the links provided on the course webpage) as a learning tool, but don't go looking for solutions.

Please include proofs with all of your answers, unless stated otherwise.

1 Basic Composition for Approximate DP

In class we proved basic composition for ϵ-DP and advanced composition for (ϵ, δ)-DP. We mentioned that basic composition also holds for (ϵ, δ)-DP, but we did not provide a proof. Prove it here.

More formally, let $A_1 : \mathcal{D} \to R_1$ be (ϵ_1, δ_1)-DP, and for $i \in \{2, \ldots, k\}$ let $A_i : \mathcal{D} \times R_1 \times \cdots \times R_{i-1} \to R_i$ be (ϵ_i, δ_i)-DP. Let $A : \mathcal{D} \to R_1 \times \cdots \times R_k$ be the composition of these algorithms, i.e., A is the algorithm that on input database D outputs $A(D) = (r_1, r_2, \ldots, r_k)$ where $r_1 = A_1(D)$ and $r_i = A_i(D, r_1, \ldots, r_{i-1})$ for all $i \in \{2, \ldots, k\}$. Prove that A is $\left(\sum_{i=1}^k \epsilon_i, \sum_{i=1}^k \delta_i\right)$-DP.

2 Parallel Composition

One extremely powerful composition lemma which is not in the book in *parallel composition*. In formally, parallel composition says that if we run DP algorithms in parallel on *disjoint* parts of the database, then there is no privacy loss whatsoever from composition.

Let X be the domain of the databases we consider, so a database D is a subset of X and the set of all databases is the power set $\mathcal{D} = 2^X$. Let $(X_1, X_2, \ldots, X_k)$ be an arbitrary partition of X. For each $i \in [k]$, let $M_i : 2^{X_i} \to R_i$ be an ϵ-DP algorithm. Let $A : \mathcal{D} \to R_1 \times \cdots \times R_k$ be the algorithm that on input D outputs $A(D) = (M_1(D \cap X_1), M_2(D \cap X_2), \ldots, M_k(D \cap X_k))$. Prove that A is ϵ-DP.

3 Peeling a Directed Graph

Let $G = (V, E)$ be a directed graph. Consider the following (non-private) algorithm. First, we compute the out-degree $d(v)$ of each node $v \in V$. We then sort V by these $d(v)$ values from smallest to largest (breaking ties arbitrarily); let σ be this sorted order. For every $v \in V$, let $d_\sigma(v)$ denote the number of edges from v to nodes *later* in σ (i.e., to nodes v' with $d(v') > d(v)$ or where $d(v') = d(v)$ but the tie-breaking put v' later than v). Output $d_\sigma(v)$ for all $v \in V$.

Now let's add privacy. For any two graph $G = (V, E)$ and $G' = (V, E')$ on the same node set, we say that G and G' are *neighboring* if $|E \setminus E'| + |E' \setminus E| = 1$, i.e., if they are different in one edge. This gives us the notion of *Edge Differential Privacy*, where we enforce differential privacy on databases that are actually graphs where the neighbor relation is being different in exactly one edge.

Design an ϵ-DP algorithm for the above problem (computing the d_σ values) in the edge DP model and prove that it is DP. You do not need to analyze its accuracy / utility, but try to add as little noise as you can.