CS 598TMC, Fall 2022
Timothy Chan

Homework 2 (due Oct 13 Thursday 5pm)

Instructions: see previous homework.

1. [*34 pts*] We are given two $n \times n$ Boolean matrices $A = (a_{ij})_{i,j\in[n]}$ and $B = (b_{ij})_{i,j\in[n]}$, and a number $\ell \le n$. We will consider variants of the Boolean matrix multiplication problem:

   (a) [*10 pts*] For every $i, j \in [n]$, we want to decide whether there is a unique index $k \in [n]$ such that $a_{ik} \wedge b_{kj}$ is true, and if true, *report this unique index* k. (The output thus consists of $O(n^2)$ indices.) Describe an $O(n^\omega)$-time algorithm to solve this problem.
   [Hint: consider $\sum_{k=1}^{n} k a_{ik} b_{kj}$.]

   (b) [*24 pts*] For every $i, j \in [n]$, we want to decide whether $|\{k \in [n] : a_{ik} \wedge b_{kj} \text{ is true}\}| \le \ell$, and if true, *report all indices k with $a_{ik} \wedge b_{kj}$ true.* (This problem generalizes part (a), which corresponds to the $\ell = 1$ case.) Describe an $\widetilde{O}(\ell^c n^\omega)$-time algorithm to solve this problem for some constant c. You may use randomization.

   [Up to 3 bonus points if you can get $c < 1$.]

   [Hint: one approach (which does not yield the best c) is to use a lemma from class: if we put k balls in k^2 bins, then with probability at least 1/2, every bin contains at most one ball.]

2. [*36 pts*] Consider the following variant of APSP, called ($\le k$)-*red APSP*: We are given an unweighted directed graph $G = (V, E)$ with n vertices, where some of the edges are labelled *red*. We are also given a number k (where $1 \le k < n$). For every pair of vertices $s, t \in V$, we want to compute the minimum length $L_k[s, t]$ over all paths from s to t that use at most k red edges.

   (a) [*12 pts*] First describe an $O(kn^2)$-time dynamic programming algorithm that solves the single-source version of the problem (i.e., for a fixed s, compute $L_k[s, t]$ for all $t \in V$).

   (b) [*24 pts*] Next describe an $\widetilde{O}(k^2 n^{2.529})$-time algorithm for ($\le k$)-red APSP.

   [Up to 3 bonus points for an $\widetilde{O}(kn^{2.529})$-time solution.]

   [*Hint*: modify Zwick's APSP algorithm. It may be helpful to compute not just $L_k[s, t]$ but $L_{k'}[s, t]$ for all $k' \le k$.]

3. [*30 pts*] Let $H(a, b)$ denote the Hamming distance between two strings a and b. Consider the following two problems:

   Problem I: Given a set A of n strings of length ℓ and another set B of n strings of length ℓ and a number Δ, decide whether there exist two strings $a \in A$ and $b \in B$ with $H(a, b) \le \Delta$, and if yes, return one such pair (a, b).

Problem II: Given a set A of n strings of length ℓ and another set B of n strings of length ℓ and a number Δ, report all strings $a \in A$ such that there exists a string $b \in B$ with $H(a, b) \leq \Delta$.

Prove that there is an algorithm for Problem I with running time $O(\ell^c n^{2-\delta})$ for some constants $c, \delta > 0$ iff there is an algorithm for Problem II with running time $O(\ell^{c'} n^{2-\delta'})$ for some constant $c', \delta' > 0$.

(It is conjectured that such an algorithm does not exist.)

[Hint: one direction is trivial. For the other direction, imitate the reduction from All-Edges Negative-Weight Triangle to Negative-Weight Triangle described in class.]