Homework 4 (due Dec 2 Wednesday 10:00am (CT))

Instructions: see previous homework.

1. [*34 pts*] Consider the following problem called *dynamic strong connectedness*: decide whether a directed graph with m edges is strongly connected (i.e., for every two vertices u and v, there exists a path from u to v and a path from v to u), under insertions and deletions of edges. You will prove a conditional lower bound for this problem.

   (a) [*17 pts*] Recall the *set intersection query* problem: build a data structure for a collection of sets $S_1, \ldots, S_\ell \subseteq [N]$ with total size $M = \sum_i |S_i|$, so that given any i and j, we can quickly enumerate all elements in $S_i \cap S_j$. In class, we have shown that if there is a data structure that could answer $\widetilde{O}(n^{3/2})$ set intersection queries with total output size $\widetilde{O}(n^{3/2})$ for an input with $M = \widetilde{O}(n^{3/2})$ and $N = \widetilde{O}(n)$ in $\widetilde{O}(n^{2-\delta})$ time for some constant $\delta > 0$, then integer 3SUM could be solved in $\widetilde{O}(n^{2-\delta'})$ time.

   Now consider the *set disjointness query* problem: build a data structure for sets $S_1, \ldots, S_\ell \subseteq [N]$ with total size $M = \sum_i |S_i|$, so that given any i and j, we can quickly decide whether $S_i \cap S_j = \emptyset$. Show that if there is a data structure that could answer $\widetilde{O}(n^{3/2})$ set disjointness queries for an input with $M = \widetilde{O}(n^{3/2})$ and $N = \widetilde{O}(n)$ in $\widetilde{O}(n^{2-\delta})$ time for some constant $\delta > 0$, then integer 3SUM could be solved in $\widetilde{O}(n^{2-\delta'})$ time for some constant $\delta' > 0$.

   Hint: create new sets $S_i \cap [0, N/2)$, $S_i \cap [N/2, N)$, $S_i \cap [0, N/4)$, etc. Queries may be given online.

   (b) [*17 pts*] Show that if there is a data structure for dynamic strong connectedness that supports edge insertions and deletions in $O(m^{1/3-\delta})$ time for some constant $\delta > 0$, then integer 3SUM could be solved in $\widetilde{O}(n^{2-\delta'})$ time for some constant $\delta' > 0$.

   Hint: to reduce set disjointness to dynamic strong connectedness, build a tripartite directed graph, and add some extra vertices and edges...

2. [*34 pts*] In class, we described an $O(n^2/\log^2 n)$-time algorithm for solving the LCS (longest common subsequence) problem when the alphabet size is constant.

   Present a slightly subquadratic algorithm for LCS that works even when the alphabet size is large. Aim for near $O(n^2/\log^2 n)$ time, ignoring $\log\log n$ factors. (Partial credit for a slower, near-$O(n^2/\log n)$-time algorithm.)

   Hint: To get near $O(n^2/\log^2 n)$, use two levels of blocking. Divide into "macro-blocks" of size w'', and divide each macro-block into "micro-blocks" of size w', for some choice of $w'' > w'$. For each pair of macro-blocks, reduce the alphabet size to $O(w'')$...

3. [*32 pts*] Consider the following variant of the 3-point collinearity problem: given a sequence of n points $p_1, \ldots, p_n$ in two dimensions, decide whether there exist i and j such that p_i, p_j, p_{i+j} lie on a common line.

   Assuming that the points have integer coordinates, describe a (randomized) algorithm that solves the problem in slightly subquadratic time. Aim for near $O(n^2/\log^2 n)$ time, ignoring $\log\log n$ factors.

   Hint: modify the algorithm from class for integer convolution-3SUM. Note that three points $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ are collinear iff $(x_2 - x_1)(y_3 - y_1) = (x_3 - x_1)(y_2 - y_1)$.