Amir Abboud, Nick Fischer

Miscellaneous 5

Barriers for Fine-Grained Reductions

Recall from yesterday's lecture that for any problem P for which we can find a nontrivially fast *verifier* (i.e., a nondeterministic and a co-nondeterministic algorithm) we cannot expect a tight hardness reduction based on SETH—at least when assuming the Nondeterministic SETH. In the next exercise we show that under this assumption we cannot prove that APSP is SETH-hard:

Problem 1 (Verifier for APSP). The goal of this exercise is to design a verifier for APSP that runs in truly subcubic time. Recall the Zero Triangle problem (from days 1 and 2 of the tutorial).

1. Give an algorithm for Zero Triangle with weights $\{-W, \dots, W\}$ that runs in time $O(n^\omega \cdot W^2)$. (This part is very similar to Problem 3 from Sheet 2.)
2. Design a verifier for Zero Triangle that runs in time $O(n^{3-\epsilon})$ for some $\epsilon > 0$.

   Hint: Follow the hash–list–count paradigm behind the fast 3SUM verifier.
3. Conclude that there is a verifier for APSP in time $O(n^{3-\epsilon})$ for some $\epsilon > 0$.

Combinatorial Boolean Matrix Multiplication

In fine-grained complexity we informally say that an algorithm is *combinatorial* if it does not use Strassen-like fast matrix multiplication. Recall that the *Combinatorial Boolean Matrix Multiplication* (BMM) hypothesis informally states that there is no $O(n^{3-\epsilon})$-time *combinatorial* algorithm for BMM. We will now prove some combinatorial lower bounds based on this hypothesis.

Problem 2 (Combinatorial Triangle Detection). Recall that in the Triangle Detection problem the goal is to decide whether a given (undirected) graph contains a triangle. Show that there is a combinatorial $O(n^{3-\epsilon})$-time algorithm for Triangle Detection (for some $\epsilon > 0$) if and only if there is a combinatorial $O(n^{3-\epsilon})$-time algorithm for BMM (for some $\epsilon > 0$), i.e., if and only if the combinatorial BMM hypothesis holds.

Hint: Revisit the subcubic equivalences presented on Tuesday.

Problem 3 (Dynamic Reachability and Bipartite Matching). Show that the following dynamic problems there is no *combinatorial* dynamic algorithm with update time $O(n^{1-\epsilon})$ (for any $\epsilon > 0$), based on the combinatorial BMM hypothesis:

1. *s-t-Reachability:* In a directed graph with designated vertices s and t that undergoes edge insertions and edge deletions, maintain whether there is a directed path from s to t.
2. *Bipartite Matching:* In an undirected bipartite graph that undergoes edge insertions and edge deletions, maintain the size of the maximum matching.

Problem 4 (Sliding-Window Hamming Distance). In the *Sliding-Window Hamming Distance* (SWHD) problem the input consists of two strings—a length-$2n$ *text* T, and a length-n *pattern* P—and the goal is to compute for all windows $i \in [n]$ the Hamming distance $\mathrm{HD}(T[i \mathinner{..} n+i], P)$.

1 Show that there is no *combinatorial* algorithm running in time $O(n^{3/2-\epsilon})$ for any $\epsilon > 0$, assuming the combinatorial BMM hypothesis.

2 Give a matching $\widetilde{O}(n^{3/2})$-time algorithm.

Hint: Encode the problem as polynomials and apply the Fast Fourier Transform.

Open Problem 5. Either strengthen the SWHD lower bound by conditioning on a more established hypothesis such as 3SUM or APSP, or give an $O(n^{3/2-\epsilon})$-time *non-combinatorial* algorithm for SWHD (for some $\epsilon > 0$).