Amir Abboud, Nick Fischer

Fine-Grained Complexity in Action 4

Over the past few days, you have learned the basic tools of fine-grained complexity. Therefore, this problem sheet will differ from the previous ones: instead of providing detailed instructions on what to reduce from, it will be up to you to identify the appropriate hypotheses—fine-grained complexity in action!

Problem 1 (Metricity). Recall that a *metric* is a function $d(x, y)$ that satisfies the axioms (i) $d(x, y) \geq 0$ (with equality if and only if $x = y$), (ii) $d(x, y) = d(y, x)$, and (iii) $d(x, y) \leq d(x, z) + d(z, y)$, for all x, y, z.

In the *Metricity* problem the input is a matrix $M \in \{0, \dots, n^c\}^{n \times n}$, and the task is to determine whether the function $d(x, y) = M[x, y]$ is a metric. Give an algorithm and a matching conditional lower bound for this problem.

Problem 2 (LCS Closest Pair). In the *Longest Common Subsequence (LCS) Closest Pair* problem the input consists of two sets of n length-d strings $X, Y \subseteq \Sigma^d$ (over some alphabet Σ of unbounded size), and the task is to find the pair $x \in X$, $y \in Y$ that maximizes $\mathrm{LCS}(x, y)$. For example, X could be a database of DNA samples from some people, and Y could be DNA samples found at crime scenes, with the goal being to determine if one of the people is linked to any unresolved crimes.

1. Convince yourself that LCS Closest Pair can be solved in time $O(n^2 \operatorname{poly}(d))$.
2. Show a conditional lower bound ruling out time $O(n^{2-\epsilon} \operatorname{poly}(d))$ (for all $\epsilon > 0$).
3. Show a conditional lower bound that even rules out computing a 100-approximation of LCS Closest Pair in time $O(n^{2-\epsilon} \operatorname{poly}(d))$ (for all $\epsilon > 0$).

Problem 3 (Sumset Size). Let $A, B \subseteq \{0, \dots, n^c\}$ be sets of size n. Their *sumset* is defined as $A + B = \{a + b : a \in A,\ b \in B\}$. What is the time complexity to compute the size $|A + B|$? Give an algorithm and a matching conditional lower bound.

Bonus Problem 4 (Longest Palindromic Subsequence). A string X is a *palindrome* if it reads the same forwards and backwards. In the *Longest Palindromic Subsequence* problem the goal is to determine a longest subsequence of a given length-n string X that is a palindrome. Give an algorithm and a matching conditional lower bound for this problem.

Bonus Problem 5 (Maximum Submatrix). In the *Maximum Submatrix* problem the input is an integer matrix $M \in \{-n^c, \dots, n^c\}^{n \times n}$, and the task is to compute the maximum entry sum of a contiguous submatrix of M:

$$\max_{1 \leq i_1 \leq i_2 \leq n} \max_{1 \leq j_1 \leq j_2 \leq n} \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} M[i, j].$$

Give an algorithm and a matching conditional lower bound for this problem.