SETH 3

Problem 1 (Longest Common Substring with Wildcards). In the classic *Longest Common Substring* problem the goal is to find the longest (consecutive) substring of two given strings $x, y \in \Sigma^n$. It is well-known that this problem can be solved in linear time $O(n)$.

In this exercise we consider the generalization where we allow a *wildcard* character $\star$ that is considered equal to all other characters. Then the *Longest Common Substring with Wildcards* problem is, given strings $x, y \in (\Sigma \cup \{\star\})^n$, to determine a longest string $z \in \Sigma^*$ that occurs as a substring of x and y.[1]

1 Optionally, convince yourself that Longest Common Substring with Wildcards can be solved in time $O(n^2)$ by a dynamic programming algorithm.

2 Show that OV $_2\!\!\longrightarrow_2$ Longest Common Substring with Wildcards.

[1] *Formally, this means that there are indices $i, j \in [n]$ so that for all $0 \leq \ell < |z|$ it holds that $x[i+\ell] \in \{z[\ell], \star\}$ and $y[j+\ell] \in \{z[\ell], \star\}$.*

Problem 2 (Single-Source Max Flow). In the *Directed Single-Source Max Flow* problem the input is a directed graph $G = (V, E)$ with edge capacities $w : E \to \mathbb{Z}_{\geq 0}$ and a designated source vertex $s \in V$, and the goal is to compute, for each sink vertex $t \in V$, the maximum s-t-flow. Show that, assuming SETH, there is no algorithm in time $O(m^{2-\epsilon})$ for any $\epsilon > 0$ (specifically, show that for graphs with $m = \widetilde{O}(n)$ edges there is no algorithm in time $O(n^{2-\epsilon})$ for any $\epsilon > 0$).

Problem 3 (NFA Acceptance). A *nondeterministic finite automaton* (NFA) is a directed graph $G = (V, E)$ where each edge is labeled with a symbol from some alphabet Σ, along with a designated *starting vertex* $v_0 \in V$ and a set of *accepting vertices* $F \subseteq V$. We say that it *accepts* a string $x \in \Sigma^*$ if there is a path $v_0\, v_1 \, \dots \, v_{|x|}$ from the starting vertex v_0 to some accepting vertex $v_{|x|} \in F$ such that the edge labels along the path equal x (i.e., such that the edge label of (v_i, v_{i+1}) equals $x[i]$).

In the *NFA Acceptance* problem the task is to decide whether a given NFA G accepts a given string x. By a standard dynamic programming algorithm this problem can be solved in time $O(m \cdot |x|)$ (where m is the number of edges in the NFA). Show that an algorithm in time $O((m \cdot |x|)^{1-\epsilon}$ for some $\epsilon > 0$ would refute SETH.

Bonus Problem 4 (Hitting Set). In the *Hitting Set* problem the input consists of n sets $S_1, \dots, S_n \subseteq [d]$ and the task is to decide if there is some $i \in [n]$ such that S_i *hits* (i.e., intersects) all other sets $S_1, \dots, S_n$. Show that Hitting Set $_2\!\!\longrightarrow_2$ OV, i.e., show that if OV can be solved in time $O(n^{2-\epsilon} \operatorname{poly}(d))$ for some $\epsilon > 0$, then Hitting Set can be solved in time $O(n^{2-\epsilon'} \operatorname{poly}(d))$ for some $\epsilon' > 0$.

Bonus Problem 5 (ETH and SETH). Recall that ETH is the hypothesis that there is some $\delta > 0$ such that 3-SAT cannot be solved in time $O(2^{\delta N})$, whereas SETH is the hypothesis that for any $\epsilon > 0$ there is some integer k such that k-SAT cannot be solved in time $O(2^{(1-\epsilon)N})$. Show that SETH implies ETH.

Hint: Try to apply the Sparsification Lemma, which we restate here for your convenience.

Lemma (Sparsification Lemma). For any $k \geq 3$ and $\epsilon > 0$, there is a constant $C = C(k, \epsilon)$ and an algorithm such that:

1. Given a k-CNF ϕ on N variables, the algorithm computes k-CNFs $\phi_1, \ldots, \phi_t$.
2. ϕ is satisfiable if and only if some ϕ_i is satisfiable.
3. $t \leq 2^{\epsilon N}$ and the algorithm runs in time $O(2^{\epsilon N} \operatorname{poly}(N))$.
4. Each ϕ_i has at most N variables and at most CN clauses.