Exercise 1

In the lecture we introduced the Orthogonal Vectors Hypothesis:

**OVH**: Given two sets $A, B \subseteq \{0, 1\}^d$ such that $|A| = |B| = n$. There is no algorithm running in time $O(n^{2-\varepsilon} \cdot \text{poly}(d))$ (for any $\varepsilon > 0$) which decides whether there exists $a \in A, b \in B$ such that $a$ and $b$ are orthogonal.

a) Consider the following variant $\text{OVH}'$ of $\text{OVH}$:

$\text{OVH}'$: Given a set $A \subseteq \{0, 1\}^d$ such that $|A| = n$. There is no algorithm running in time $O(n^{2-\varepsilon} \cdot \text{poly}(d))$ (for any $\varepsilon > 0$) which decides whether there exist $a, a' \in A$ such that $a$ and $a'$ are orthogonal.

Prove that $\text{OVH}'$ and $\text{OVH}$ are equivalent.

b) Consider the problem of finding the maximum inner product of elements of two sets:

$\text{MaxInnerProduct}$: Given two sets $A, B \subseteq \mathbb{R}^d_{\geq 0}$ such that $|A| = |B| = n$, compute the maximum

$$\max \{ \langle a, b \rangle \mid a \in A, b \in B \},$$

where $\langle \cdot, \cdot \rangle$ denotes the standard inner product of $\mathbb{R}^d_{\geq 0}$.

Prove that there is no algorithm running in time $O(n^{2-\varepsilon} \cdot \text{poly}(d))$ (for any $\varepsilon > 0$) for this problem unless $\text{OVH}$ fails.
Exercise 2

From your algorithms classes you may know the problem of finding a string $P$ (often called pattern) in another string $T$ (often called text). This well-known problem is often called Pattern Matching; there are algorithms for this problem that run in time $O(|P| + |T|)$.

Instead of finding a single pattern string $P$, we are now interested in finding any substring of $T$ that can be generated by a given regular expression. Formally, consider the following problem:

**RegExPatternMatching**: Given a regular expression $R$ of size $m$, and a text $T$ of size $n$, determine if any substring $P$ of $T$ can be derived from $R$.

a) Prove that there is no algorithm running in time $O((mn)^{1-\epsilon})$ (for any $\epsilon > 0$) for RegExPatternMatching unless OVH fails.

As it turns out, for specific classes of regular expressions, there are faster algorithms to solve this problem. Consider homogeneous regular expressions:

A regular expression $R$ is called homogeneous of type “$o_1 o_2 \ldots o_l$” (where $o_i \in \{\circ, \ast, +, |\}$) if there exist $a_1, \ldots, a_p$, characters or homogeneous regular expressions of type $o_2 \ldots o_1$, such that $R = o_1(a_1, \ldots, a_p)$.

For example, the regular expression $[(a \circ b \circ c) \mid b \mid (a \circ b)]^*$ is homogeneous of type “$\ast | \circ$”, the regular expression $(a^*) \mid (b^+)$ is not homogeneous.

b) Give an $O(m + n)$ time algorithm for RegExPatternMatching where the regular expression is homogeneous of type “$\circ$” or of type “$\ast \circ$”.

c) Prove that there is no $O((mn)^{1-\epsilon})$ algorithm (for any $\epsilon > 0$) for RegExPatternMatching where the regular expression is homogeneous of type “$| \circ |$” unless OVH fails.

Prove the same result for homogeneous regular expressions of type “$| \circ \ast$”.

*) (Bonus) Prove the result from c) for homogeneous regular expressions of type “$\circ \ast$”.

---

1See for example Knuth, Morris, and Pratt’s algorithm.