Fine-grained Parameterized Algorithms

FPA · SoSe-2024 · `tcs.uni-frankfurt.de/parameterized/` · 2024-07-03 · *2d88037*

Problem set 9: OVH and ETH

9.1 Fingerübungen. Let $0 < \varepsilon < 0.5$ and $c \in \mathbb{N}$ be arbitrary. Consider the following functions:

$$n^{2-\varepsilon}, \quad n \cdot \log^c n, \quad \frac{n^2 \log\log n}{\log n}, \quad \frac{n^2}{2^{\sqrt{n}}}, \quad n^2, \quad n^{1+\varepsilon}, \quad n^{2-\frac{1}{\log\log n}}.$$

a) Arrange these functions in increasing order of growth rate.
b) Which of these functions are contained in $n^{1+o(1)}$ and $n^{2-o(1)}$, respectively?

‼ **Skill-9a.** **OVH:** *I can use the orthogonal vectors hypothesis (OVH) to prove conditional lower bounds for polynomial-time problems.*

9.2 OVH Variant. Recall the Orthogonal Vectors Hypothesis (OVH):

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

Consider the following variant OVH′ of OVH:

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

Prove that OVH′ and OVH are equivalent.

9.3 Maximum Inner Product. Consider the problem of finding the maximum inner product of elements of two sets:

> **MaxInnerProduct**: Given two sets $A, B \subseteq \mathbb{R}^d_{\geq 0}$ with $|A| = |B| = n$, 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 $n^{2-\epsilon} \cdot \text{poly}(d)$ (for any $\epsilon > 0$) for this problem unless OVH fails. *Hint:* You may want to use 9.2.

9.4 Regular Expressions. 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 there is a substring P of T that can be derived from R.

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

‼ <u>**Skill-9b.**</u> **ETH:** *I can use the Exponential Time Hypothesis (ETH) and the Strong Exponential Time Hypothesis (SETH) to prove conditional lower bounds for problems in polynomial time or in exponential time.*

9.5 SETH⇒LDOVH. Consider the following stronger variant of the Orthogonal Vectors Hypothesis (OVH), called LDOVH (Low-dimensional OVH): For all $\epsilon > 0$, there is a constant $c > 0$, such that the Orthogonal Vectors problem cannot be solved in time $O(n^{2-\epsilon})$, even when the input is restricted restricted to $d \leq c \cdot \log n$. Prove that SETH implies LDOVH. *Hint:* *You may use the Sparsification Lemma.*

9.6 Subsetting Sum Fun. A classical NP-hard problem is the Subset Sum problem:

> **Subset Sum**: Given a set of n distinct integers $X = \{1 \leq x_1 < \cdots < x_n\}$ and an integer t, determine whether there is a subset $A \subseteq X$ that sums up to t, that is $\sum_{a \in A} a = t$.

Assuming the Exponential Time Hypothesis (ETH), prove that there is a $\delta > 0$, such that no algorithm for Subset Sum has a running time of $O^*(2^{\delta n})$.