Karl Bringmann and Marvin Künnemann **Summer 2019**

Fine-Grained Complexity Theory, Exercise Sheet Zero

`www.mpi-inf.mpg.de/departments/algorithms-complexity/teaching/summer19/fine-complexity/`

Please subscribe to our mailing list under

https://lists.mpi-inf.mpg.de/listinfo/finegrained19

if you haven't done so yet. Further, if you plan to take the exam, please write an email (using the mail registered on the mailing list) containing your name and matriculation number to Philip (wellnitz@mpi-inf.mpg.de).

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 **OVH′** of **OVH**:

   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 **OVH′** and **OVH** are equivalent.

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

   MaxInnerProduct: Given two sets $A, B \subseteq \mathbb{R}^d_{\geq 0}$ 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 **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 $O(|P| + |T|)$[1].

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 $O((mn)^{1-\varepsilon})$ (for any $\varepsilon > 0$) 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 "$o_1 o_2 \dots o_l$"* (where $o_i \in \{\circ, *, +, |\}$) if there exist $a_1, \dots, a_p$, characters or homogeneous regular expressions of type $o_2 \dots o_l$, such that $R = o_1(a_1, \dots, a_p)$.

For example, the regular expression $[(a \circ b \circ c) \,|\, b \,|\, (a \circ b)]^*$ is homogeneous of type "$* \,|\, \circ$", the regular expression $(a^*) \,|\, (b^+)$ 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 "$* \circ$".

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

$\star$) *(Bonus)* Prove the result from c) for homogeneous regular expressions of type "$\circ *$".

[1] See for example Knuth, Morris, and Pratt's algorithm.