Karl Bringmann and Marvin Künnemann **Summer 2019**

Fine-Grained Complexity Theory, Exercise Sheet 1

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

Total Points: **40** + 2 bonus points   Due: Tuesday, **April 30**, 2019

You are allowed to collaborate on the exercise sheets, but you have to write down a solution on your own, ***using your own words****. Please indicate the names of your collaborators for each exercise you solve. Further, cite all external sources that you use (books, websites, research papers, etc.).*

You need to collect at least 50% of all points on exercise sheets to be admitted for the exam.

Exercise ◯◯ — 2 bonus points

Read the lecture notes (of the last three lectures), identify as many typos and other mistakes as you can, and add them as a list to your solutions. You get one bonus point for at least one typo/mistake and 2 bonus points for at least five typos/mistakes.

Exercise 1 — 8 + 2 points

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

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

a) Order these functions according to their asymptotic growth. Justify your answers.

b) Which of these functions are contained in $n^{1+o(1)}$ and $n^{2-o(1)}$, respectively?

Exercise 2 — 5 + 5 points

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 $O(n^{2-\varepsilon} \cdot \mathrm{poly}(d))$ (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 **UOVH** of **OVH**:

   UOVH (Unbalanced **OVH**): Given two sets $A, B \subseteq \{0,1\}^d$ such that $|A| = n, |B| = \sqrt{n}$. There is no algorithm running in time $O(n^{1.5-\varepsilon} \cdot \mathrm{poly}(d))$ (for any $\varepsilon > 0$) which decides whether there exist $a \in A, b \in B$ such that a and b are orthogonal.

   Prove that **UOVH** and **OVH** are equivalent.

b) Consider the following stronger variant **LDOVH** of **OVH**:

   LDOVH (Low-dimensional **OVH**): For all $\varepsilon > 0$ there is a (constant) $c > 0$ such that the Orthogonal Vectors problem cannot be solved in time $O(n^{2-\varepsilon})$ even restricted to $d \le c \cdot \log n$.

   Prove that **SETH** implies **LDOVH**.

   (Hint: You may use the Sparsification Lemma from the lecture.)

Exercise 3 — 10 points

One of the classical string problems is finding the longest common substring of two given strings A and B. In this exercise we will consider a variation of that problem, where we allow one string to contain *wild card* characters, that is, characters that are considered equal to any other character. Formally, consider the following problem:

Longest Common Substring With Don't Cares: Given a string A of length n over some alphabet Σ and string B of length n over the alphabet $\Sigma \cup \{\star\}$, find the length of the longest string that is a substring of both A and B, where a "$\star$" in B can be treated as any character from the alphabet Σ.

Prove that there is no $O(n^{2-\varepsilon})$ time algorithm (for any $\varepsilon > 0$) for this problem unless **OVH** fails.

Exercise 4 — 10 points

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 $\Sigma_{a \in A} a = t$.

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