max planck institut informatik

SAARLAND UNIVERSITY

COMPUTER SCIENCE

Karl Bringmann and Sebastian Krinninger **Summer 2016**

Exercises for Complexity Theory of Polynomial-Time Problems

https://www.mpi-inf.mpg.de/departments/algorithms-complexity/teaching/summer16/poly-complexity/

Exercise sheet 1 Due: **Monday, May 9, 2016**

Total points : 40

Normally homework is to be handed in on Thursday in the lecture. Since Thursday May 5 is a holiday, this time you should either email an electronic version of your assignment submission to gjindal@mpi-inf.mpg.de or give it to Gorav in his office at Room 425, Building E1.3. If Gorav is not in his office then you can just slide your submission under the door of his office.

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.

Exercise 1 (*10 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 $O(n^{2-\epsilon} \cdot \text{poly}(d))$ (for any $\epsilon > 0$) which decides whether there exists $a \in A, b \in B$ such that a and b are orthogonal.

a) (5 points) 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 $O(n^{2-\epsilon} \cdot \text{poly}(d))$ (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.

b) (5 points) 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 $O(n^{1.5-\epsilon} \cdot \text{poly}(d))$ (for any $\epsilon > 0$) 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.

Exercise 2 (*13 points*) Consider the following problem:

2SAT+2Clauses: Given a 2-CNF formula on n variables and m clauses, along with two additional clauses of arbitrary length. Decide whether the combined formula is satisfiable.

a) (7 points) Show that if **2SAT+2Clauses** can be solved in time $O((m+n)^{2-\epsilon})$ then **OVH** fails.

b) (6 points) Show that **2SAT+2Clauses** can be solved in time $O(mn + n^2)$.
Hint*: Construct a suitable graph with all the literals (and their negations) as vertices. Since each clause from 2SAT contains two literals, each clause should somehow correspond to some edge of this graph. Now find a graph property which is equivalent to satisfiability of the given 2SAT+2Clauses formula.*

Exercise 3 (*7 points*) Consider the Orthogonal Vectors problem with $d = n$. This problem can be trivially solved in time $O(n^3)$. Prove that this case of the Orthogonal Vectors problem can be solved in time $O(n^\omega)$ where ω is the exponent of matrix multiplication, i.e, two $n \times n$ integer matrices can be multiplied[1] in $O(n^\omega)$. It is known that $\omega < 2.372873$.

Exercise 4 (*10 points*) Consider the following stronger variant **LDOVH** of **OVH**:

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 restricted to $d \leq c \cdot \log n$.

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

You can use the following fact to solve this exercise. You do not need to proof the Sparsification Lemma, just assume it to be true.

Lemma (Sparsification Lemma)**.** *For all $\epsilon > 0$ and positive k, there is a constant $C = C(\epsilon, k)$ such that any k-CNF formula φ with n variables and m clauses can be expressed as $\varphi = \bigvee_{i=1}^{t} \psi_i$, where $t \leq 2^{\epsilon n}$ and each ψ_i is a k-CNF formula with the same variable set as φ and at most Cn clauses. Moreover, this disjunction can be computed by an algorithm running in time $O(\mathrm{poly}(n, m) \cdot 2^{\epsilon n})$.*

Hint *: Use the Sparsification Lemma to make the number of clauses $O(n)$ and then use the proof from the lecture that **SETH** implies **OVH**.*

[1] Here we mean the standard $(+, \cdot)$ matrix multiplication that you learned in school, not the Boolean matrix multiplication discussed in the first lecture.