NTIN 086 Selected Topics from Computational Complexity I winter 2018/2019

First homework assignment

Turn-in by January 6, 2019

Problem 1. *SAT-Solver.* Design a deterministic algorithm that for an input CNF formula of size n with k clauses decides whether it is satisfiable in time $2^k \cdot poly(n)$. (*Hint:* Compute the number of satisfying assignments by computing the number of satisfying assignments when we leave out one of the clauses and when we set all the literals in that clause to false.)

Problem 2. *Colors of Ostrava.*

a) Let OV_d be the problem: *Given n vectors in $\{0,1\}^d$, is there an orthogonal pair?*

Let Color-OV_d be the problem: *Given n black vectors and n white vectors in $\{0,1\}^d$, is there an orthogonal black-white pair?*

Prove that for every d, if OV_d can be solved in time $t(n)$ then Color-OV_{d-2} can be solved in time $O(t(n))$.

b) Let $3SUM$ be the problem: *Given n numbers, are there three which sum to zero?*

Let Color-$3SUM$ be the problem: *Given n numbers, where each number is colored either red, green, or blue, is there a red number r, a blue number b, and a green number g which sum to zero?*

Prove that if $3SUM$ can be solved in time $t(n)$ then Color-$3SUM$ can be solved in time $O(t(n))$.

Problem 3. *Search vs Decision.*

a) Let Search-OV_d be the problem: *Given n vectors in $\{0,1\}^d$ find all the vectors that are orthogonal to some other vector in the set.*

Prove that for every d, if OV_d can be solved in time $O(n^{2-\epsilon})$, for some $\epsilon > 0$, then Search-OV_d can be solved in time $O(n^{2-\delta})$, for some $\delta > 0$.

b) Let Search-$3SUM$ be the problem: *Given a set S of n numbers, find the subset $S' \subseteq S$ such that for each $s \in S'$, there are $a, b \in S$ such that $a + b + c = 0$.*

Prove that if $3SUM$ can be solved in time $O(n^{2-\epsilon})$, for some $\epsilon > 0$, then Search-$3SUM$ can be solved in time $O(n^{2-\delta})$, for some $\delta > 0$.

Problem 4. *k-Subgraph.* Let k-subgraph problem be the problem: *Given a graph G on n vertices and H on k vertices, is there a copy of G in H?*

Show that if we can solve k-subgraph problem in time $f(k) \cdot n^c$, for some function f and $c \geq 2$, then we can actually output a copy of H in G, if it exists, in time $g(k) \cdot n^c$, for some function g. Make $g(k)$ as small as possible relative to $f(k)$.

Problem 5. *Transitive closure.* Show that the transitive closure of a graph on n vertices can be found in time $O(n^\omega)$, where ω is the exponent of matrix multiplication. (*Hint:* Partition the vertices of the input graph G in two parts and try to solve the problem separately in each part.)

Problem 6. $(\min, +)$-*Matrix product.* Show that if $(\min, +)$-matrix product can be solved in time $T(n)$ then APSP (All-Pairs-Shortest-Path in graphs with non-negative weights) can be solved in time $O(T(n))$.