Homework 6. Due June 8th, 10PM

CS180: Algorithms and Complexity
Spring 2018

GUIDELINES:

- Upload your assignments to Gradescope by 9:59 PM.
- Follow the instructions mentioned on the course webpage for uploading to Gradescope very carefully (including starting each problem on a new page and matching the pages with the assignments); this makes it easy and smooth for everyone. As the guidelines are simple enough, bad uploads will not be graded.
- You may use results and algorithms from class without proofs or too many details as long as you state what you are using clearly.
- Most importantly, make sure you adhere to the policies for academic honesty set out on the course webpage. The policies will be enforced strictly. Homework is a stepping stone for exams; keep in mind that reasonable partial credit will be awarded and trying the problems will help you a lot for the exams.

1. Consider the problem FIND-IS defined as follows: "Given a graph G and a number k as input, find an independent set of size k in G if one exists." Recall the INDEPENDENT-SET decision problem from class: "Given a graph G and a number k, does G contain an independent set of size at least k?". Give a polynomial-time reduction from FIND-IS to INDEPENDENT-SET. [.75 points]

2. Give a polynomial-time Karp reduction from 3SAT to Vertex-Cover. For full-credit for this problem, you should not use transitivity but have to show the three steps as we did in class for the reduction from 3SAT to Independent-Set. [.75 points]

3. Consider the problem LPS defined as follows: "Given a matrix $A \in \mathbb{R}^{n\times n}$, a vector $b \in \mathbb{R}^n$ and an integer $k > 0$, does there exist a vector $x \in \mathbb{R}^n$ with at most k non-zero entries such that $A \cdot x \geq b$". Here $A \cdot x$ denotes the usual matrix-vector product and for two vectors u, v, we say $u \geq v$ if for every i, $u_i \geq v_i$. Give a polynomial-time reduction from 3SAT to LPS. [.75 points]
(Hint: Use the reductions done in class along-with transitivity of $\leq_P$ to first pick the "right" starting point and then design a reduction from this starting point.)

4. Show that the following problems are in NP.

(a) Given two graphs $G = (V, E)$ and $H = (W, F)$, we say G, H are *isomorphic* if there exists a bijection $\sigma : V \to W$ such that for any two $u, v \in V$, $\{u, v\} \in E$ if and only if $\{\sigma(u), \sigma(v)\} \in F$ (i.e., two vertices $u, v \in V$ are adjacent if and only if $\{\sigma(u), \sigma(v)\}$ are adjacent).

Let GraphIsomorphism $= \{(G, H) :$ G,H are isomorphic graphs$\}$. Show that GraphIsomorphism is in NP.

(b) Given a graph $G = (V, E)$, a subset of vertices I is a dominating set if every vertex in the graph is either in I or adjacent to a vertex in I. Let DominatingSet $= \{(G, k) :$ G has a dominating set of size $k\}$. Show that DominatingSet is in NP.

Additional Problems. DO NOT turn in answers for the following problems - they are meant for your curiosity and understanding.

1* Given a graph $G = (V, E)$, a valid 3-coloring of a G is a mapping $\chi : V \to \{1, 2, 3\}$ such that for any adjacent vertices u, v, $\chi(u) \neq \chi(v)$. Consider the problem Find-3Coloring defined as follows: "Given a graph G, find a valid 3-coloring of G if one exists." and the 3Coloring decision problem from class: "Given a graph G, is there a valid 3-coloring of G". Give a polynomial-time reduction from Find-3Coloring to 3Coloring. You don't have to prove correctness or analyze time-complexity.

[Hint: Note that if $|V| \geq 4$, then two vertices must get the same color in any proper 3-coloring. Use the black-box access to 3Coloring to find two vertices that can get the same color and use this to decrease the size of the graph you are working with and repeat.]

2 For this problem we need the notion of multi-variate polynomials over variables $x_1, \ldots, x_n$ and how they are specified. To review some terminology, we say a *monomial* is a product of a real-number co-efficient c and each variable x_i raised to some non-negative integer power a_i: we can write this as $cx_1^{a_1}x_2^{a_2}\cdots x_n^{a_n}$. A polynomial is then a sum of a finite set of monomials. (For example, $2x_1^2x_2x_3^4 + x_1x_3 - 6x_2^2x_3^2$ is a polynomial.)

We say a polynomial P is of degree at most d, if for any monomial $cx_1^{a_1}x_2^{a_2}\cdots x_n^{a_n}$ appearing in P, $a_1 + a_2 + \cdots a_n \leq d$. (For example, the degree of the previous polynomial is 7). One can represent a n-variable polynomial of degree d by at most $(n+1)^d$ numbers (by d-dimensional arrays for instance - but let us ignore the actual details of the representation as this is not important and any representation will work for us).

Consider the problem Poly-Root defined as follows: "Given a polynomial with integer coefficients of degree at most 6 as input, decide if there exists a $x \in \mathbb{R}^n$ such that $P(x) = 0$." Show that 3SAT is polynomially easier than Poly-Root. You don't have to write down the coefficients of the polynomials explicitly in your reduction - you can leave them as summations if it is more convenient for you (one reason why we didn't specify the exact representation).

Hint: Try to define a polynomial equation for each clause in your 3SAT instance along with some other equations for variables and combine them into one big polynomial by using the fact that for any set of numbers $(a_1, \ldots, a_m)$, $\sum_i a_i^2 = 0$ if and only if $a_i = 0$ for all $i \in [m]$.

3 Chapter 8, Problems 1,3, 6, 26, 27, 30, 31, 41 from [KT]

4 Problems 8.3, 8.4, 8.6, 8.8, 8.14, 8.23 from Chapter 8 of [DPV].