Practical sessions for Introduction to Complexity and Computability - 7

January 7th, 2020

Exercises

1. (previous HW) Find polynomial reductions between CLIQUE, VC, and IND where VC (vertex cover) and IND (independent set) are the following problems.

   Instance (the same for VC and IND): A graph G and an integer $k \geq 1$.

   Question (VC problem): Does G contain a vertex cover of size k? (A *vertex cover* of a graph G is a set of vertices in G that are incident to all edges of G.)

   Question (IND problem): Does G contain an independent set of size k? (An *independent set* in a graph G is a set of vertices in G with no edges between them.)

2. (previous HW) Show that the problems DOM and CC are **NP**-complete by polynomial reduction of VC.

   Instance (DOM): A graph G and an integer $k \geq 1$.

   Question (DOM): Does G contain a dominating set of size k? (A set D of vertices in a graph G is *dominating* if every vertex of G that is not in D has a neighbor in D.)

   Instance (CC): A directed graph G and an integer $k \geq 1$.

   Question (CC): Does G contain a cycle cover of size k? (A *cycle cover* of G is a set of vertices containing at least one vertex from each cycle in G.)

3. (previous HW) We know that the following problem HC (Hamiltonian cycle) is **NP**-complete.

   Instance (HC): An (undirected) graph G.

   Question (HC): Does G have a Hamiltonian cycle, i.e. a cycle through all the vertices?

   Show that also the following problems DHC, HP, HP+ are **NP**-complete.s

   Instance (DHC): A directed graph G.

   Question (DHC): Does G have a directed Hamiltonian cycle?

   Instance (HP): A (undirected) graph G.

   Question (HP): Does G have a Hamiltonian path, i.e. a path through all the vertices?

   Instance (HP+): A (undirected) graph G and two vertices s, t.

   Question (HP+): Does G have a Hamiltonian path between s and t?

4. Show that the following problems IP (integer programming) and BIP (binary integer programming) are **NP**-hard by polynomial reductions of some of the above mentioned problems.

   Instance (the same for IP and BIP): An $m \times n$ matrix A over integers and a vector b of m integers.

   Question (IP): Is there a vector x of n integers such that $Ax \geq b$?

   Question (BIP): Is there a vector $x \in \{0,1\}^n$ such that $Ax \geq b$?

5. We know that the following problem PAR (partition) is **NP**-complete.

   Instance (PAR): A set A of n items with associated size $s(a) \in \mathbb{N}$ for every $a \in A$.

   Question (PAR): Can A be partitioned into two subsets of equal total size (i.e. $\frac{1}{2}\sum_{a\in A} s(a)$)?

   Show that also the following problems 3-PAR (3-partition), KS (knapsack), SCH (scheduling) are **NP**-complete.

   Instance (3-PAR): A set A of n items with associated size $s(a) \in \mathbb{N}$ for every $a \in A$.

   Question (3-PAR): Can A be partitioned into 3 subsets of equal total size (i.e. $\frac{1}{3}\sum_{a\in A} s(a)$)?

Instance (KS): A set A of n items with associated size $s(a)$ and value $v(a)$ for every $a \in A$, a size of knapsack k, and a target value w, all natural numbers.

Question (KS): Is there $A' \subseteq A$ with $\sum_{a \in A'} s(a) \leq k$ and $\sum_{a \in A'} v(a) \geq w$?

Instance (SCH): A set A of n tasks with associated running time $t(a)$ for every $a \in A$, and $m, D \in \mathbb{N}$.

Question (SCH): Can A be scheduled to m processors in parallel running time $\leq D$?

6. In the optimization version of the vertex cover problem (Min-VC) we ask for a vertex cover of minimal size for a given graph G. Consider the following (greedy) approximation algorithm.

   For an arbitrary edge uv of G we add both u, v to a set S (initially empty) and we remove all edges incident to u or v from G. We repeat this step until G is empty. Then we output S as a vertex cover of G.

   Determine the approximation ratio of this algorithm.

7. In the Max-SAT problem we ask for an assignment of a given CNF formula that satisfies maximal number of clauses. Show that there is a 2-approximation algorithm running in polynomial time for this problem.

8. We know that every planar graph can be colored by 4 colors. However, the problem of deciding whether a given planar graph can be colored with 3 colors is NP-complete. In the optimization version we ask for the chromatic number of a given planar graph. (The *chromatic number* of a graph G is the least number of colors in a proper coloring of G, i.e. coloring assigning distinct colors to each pair of adjacent vertices.)

   Find an approximation algorithm running in polynomial time for this problem with approximation ratio less than $3/2$.