CS331: Algorithms and Complexity
Homework VI

Trung Dang   Ryan Park   Kevin Tian

Due date: December 9, 2024, end of day (11:59 PM), uploaded to Canvas.

Late policy: 15% off if submitted late, and 15% off for every further 24 hours before submission.

Please list all collaborators on the first page of your solutions.

When runtimes are unspecified, slower runtimes than the intended solution receive partial credit.

1 Problem 1

Let L and M be two decision problems in NP.

(i) **(10 points)** Let $L \cap M$ be the decision problem where an input $x \in \{0,1\}^*$ is in $L \cap M$ iff it is in both L and M. Is $L \cap M$ in NP? If so, prove it, and otherwise, give a counterexample.

(ii) **(10 points)** Let $L \cup M$ be the decision problem where an input $x \in \{0,1\}^*$ is in $L \cup M$ iff it is in either L or M. Is $L \cup M$ in NP? If so, prove it, and otherwise, give a counterexample.

2 Problem 2

Prove that the following decision problems are coNP-complete.

(i) **(10 points)** UnSAT defined as: $x \in \mathsf{UnSAT}$ iff $x = \langle \Phi \rangle$ encodes an unsatisfiable Boolean formula Φ, i.e., no variable assignment causes Φ to evaluate to **True**.

(ii) **(10 points)** $\mathsf{Tautology}$ defined as: $x \in \mathsf{Tautology}$ iff $x = \langle \Phi \rangle$ encodes a *tautology* Φ, i.e., a Boolean formula Φ that evaluates to **True** for all variable assignments.

3 Problem 3

We say that two unweighted graphs $G = (V, E)$ and $H = (U, F)$ are *isomorphic* iff $|V| = |U|$, there is a surjective function $\pi : V \to U$ (i.e., every $u \in U$ has some $v \in V$ such that $\pi(v) = u$), and $(v, v') \in E$ iff $(\pi(v), \pi(v')) \in F$. Intuitively, this means that for $n = |V| = |U|$ there is a way to label the vertices of V and U with the numbers 1 through n in a consistent way, such that the presence or absence of each edge $(i, j) \in [n] \times [n]$ is consistent in both graphs.

(i) **(10 points)** Consider the decision problem GI ("graph isomorphism") defined as: $x \in \mathsf{GI}$ iff $x = \langle G, H \rangle$ encodes a pair of isomorphic unweighted graphs G, H. Prove that $\mathsf{GI} \in$ NP.

(ii) **(10 points)** Consider the decision problem SGI ("subgraph isomorphism") defined as: $x \in \mathsf{SGI}$ iff $x = \langle G, H \rangle$ encodes a pair of unweighted graphs G, H, such that there is an induced subgraph[1] G' of G that is isomorphic to H. Prove that SGI is NP-complete.

[1]Recall from Section 4.1, Part I that we say $G' = (V', E')$ is an induced subgraph of $G = (V, E)$ if $V' \subseteq V$, and $(u, v) \in E'$ iff $(u, v) \in E$ for all $(u, v) \in V' \times V'$. In other words, G' is obtained from G by deleting some of its vertices (but not deleting any edges between any non-deleted vertices).

4 Problem 4

(i) **(10 points)** Consider the decision problem Partition defined as: $x \in$ Partition iff $x = \langle A \rangle$ encodes A, a length-n Array instance of positive integers $\{a_i\}_{i\in[n]}$, such that there is a subset indexed by $S \subseteq [n]$, such that the total values of S and $[n] \setminus S$ are equal:

$$\sum_{i\in S} a_i = \sum_{i\in[n]\setminus S} a_i.$$

Prove that Partition is NP-complete.

(ii) **(5 points)** Consider the decision problem Knapsack defined as: $x \in$ Knapsack iff $x = \langle W, V, B, T \rangle$ encodes W and V, two length-n Array instances of positive integer weights $\{w_i\}_{i\in[n]}$ and values $\{v_i\}_{i\in[n]}$ respectively, a weight budget $B \in \mathbb{N}$, and a target value $T \in \mathbb{N}$, such that there exists a subset of items $S \subseteq [n]$ satisfying

$$\sum_{i\in S} w_i \le B \text{ and } \sum_{i\in S} v_i \ge T.$$

In other words, an input $\langle W, V, B, T \rangle$ is in Knapsack iff it is possible to take a subset of items with total weight at most B and total value at least T. Prove that Knapsack is NP-complete.

(iii) **(5 points)** In Section 3.3, Part II, we gave a zero-one knapsack algorithm using $O(nB)$ time. Even if P $\neq$ NP, why does this not contradict the NP-completeness of Knapsack?

5 Problem 5

Let $\{x_j\}_{j\in[n]}$ be Boolean variables, and let $\Phi = \bigwedge_{i\in[m]} \phi_i$ be a 2CNF formula, i.e., a CNF formula where each clause ϕ_i is the disjunction ("**or**") of two literals $\ell \vee \ell'$, where $\ell, \ell' \in \{x_j, \neg x_j\}_{j\in[n]}$.

(i) **(5 points)** Construct an *implication graph* $G = (V, E)$ based on Φ with $2n$ vertices and $2m$ edges as follows. Each literal $\ell \in \{x_j, \neg x_j\}_{j\in[n]}$ has an associated vertex in V. Further, for each clause $(\ell \vee \ell')$, we add two edges from $\neg\ell$ to ℓ' and from $\neg\ell'$ to ℓ.

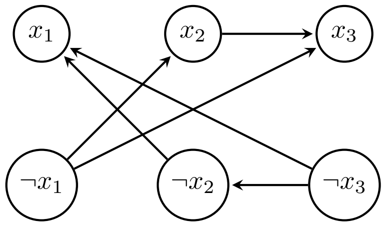

Figure 1: Implication graph for the 2CNF formula $(x_1 \vee x_2) \wedge (x_1 \vee x_3) \wedge (\neg x_2 \vee x_3)$.

Prove that $\{x_j, \neg x_j\}_{j\in[n]} \in \{\textbf{True}, \textbf{False}\}^{2n}$ is a satisfying assignment to Φ iff each pair of contradictory literals $\{x_j, \neg x_j\}$ are set to $\{\textbf{True}, \textbf{False}\}$ in some order, and whenever vertex ℓ can reach vertex ℓ' in G, then $\ell = \textbf{True}$ implies $\ell' = \textbf{True}$ as well.

(ii) **(10 points)** Prove that Φ is satisfiable iff no pair of contradictory literals $\{x_j, \neg x_j\}$ lie in the same strongly connected component (SCC) in G.

(iii) **(5 points)** Conclude that there is a poly(m, n)-time algorithm that determines whether or not a 2CNF formula with n variables and m clauses is satisfiable (i.e., the 2SAT problem).

6 Problem 6

(10 bonus points) Complete the course evaluation for CS 331 on Canvas, and attach any sort of confirmation (e.g., a clipped screenshot) to your submission PDF.