Computing Theory

Midterm Examination on May 16, 2019
Spring Semester, 2019

Problem 1 (20 points) The problem DOMINATING SET asks, given an undirected graph $G = (V, E)$ and a goal k, if there exists a set $D \subseteq V$ with at most k nodes such that every node not in D is adjacent to at least one element of D. Prove that DOMINATING SET is NP-complete. (Hint: Recall that the NP-complete problem NODE COVER asks, given an undirected graph $G = (V, E)$ and a goal k, if there exists a set $C \subseteq V$ with at most k nodes such that each edge of G has at least one of endpoints in C.)

Proof: It is clear that DOMINATING SET is in NP: guess a set with at most k nodes and verify that it is a dominating set of the graph. Given an instance $(G(V, E), k)$ of NODE COVER, we transform it to an instance $(G'(V', E'), k)$ of DOMINATING SET as follows. For each edge $(u, v) \in E$, we add a new node $w_{u,v}$ that is connected to both u and v. So $V' = V \cup \{w_{u,v} \mid (u, v) \in E\}$ and $E' = E \cup \{(u, w_{u,v}), (w_{u,v}, v) \mid (u, v) \in E\}$. It is clear that the reduction runs in polynomial time. We now prove that G has a node cover of at most size k if and only if G' has a dominating set of at most size k.

> ($\rightarrow$): If there exists a node cover set C of at most size k in G, then C is also a dominating set of G'.
>
> ($\leftarrow$): Suppose that there exists a dominating set D of at most size k in G'. For all node $w_{u,v} \in D$ corresponding to some edge $(u, v) \in E$, we replace $w_{u,v}$ of D by u to produce D'. Note that if u is already in D, we just remove $w_{u,v}$. By removing $w_{u,v}$ from D, the only nodes that might become uncovered are $w_{u,v}$, u, and v, but they are covered by u. Clearly, D' is a node cover of G.

■

Problem 2 (20 points) The problem PARTITION asks, given a set S of integers, if there exists a partition of S into two subsets S_1 and $S_2 = S - S_1$ such that $\sum_{x \in S_1} x = \sum_{x \in S_2} x$. Prove that PARTITION is NP-complete. (Hint: Recall that the NP-complete problem SUBSET SUM asks, given a set X of integers and a goal k, if there exists a subset $Y \subseteq X$ adding up to exactly k.)

Proof: It is clear that PARTITION is in NP: guess a subset S_1 of S and verify that whether $\sum\limits_{x \in S_1} x = \sum\limits_{x \in S_2} x$. We now reduce SUBSET SUM to PARTITION. The reduction is $S = X \cup \{t - 2k\}$, where t is the sum of members of X. It is clear that the reduction runs in polynomial time. We now prove that $(X, k) \in$ SUBSET SUM if and only if $S \in$ PARTITION.

> ($\rightarrow$): If there exists a subset $Y \subseteq X$ adding up to k, then the remaining members in X adding up to $t-k$. Therefore, there exists a partition of X' into $X_1 = Y \cup \{t-2k\}$ and $X_2 = X' - X_1$ such that each partition sums to $t - k$.
>
> ($\leftarrow$): If there exists a partition of X' into two sets X_1 and X_2 such that each partition sums to $t - k$, then a set of numbers adding up to $t - k$ is obtained by removing this number from one of two sets which contains the number $t - 2k$.

■

Problem 3 (20 points) The problem UNREACHABILITY asks, given an undirected graph $G = (V, E)$, two nodes a and b, and a goal k, if there *does not exist* a simple path of length at least k from node a to b. Prove that UNREACHABILITY is coNP-complete.

Proof: Recall that L is NP-complete if and only if its complemet $\bar{L} = \Sigma^* - L$ is coNP-complete. The problem REACHABILITY (L) askes, given an undirected graph $G = (V, E)$, two nodes a and b, and a goal k, if there *exists* a simple path of length at least k from node a to b. Thus we only need to prove that REACHABILITY (L) is NP-complete. It is clear that REACHABILITY (L) is in NP: guess a simple path of length at least k from node a to b and verify it. Recall that HAMILTONIAN PATH is NP-complete. Clearly, there exists a Hamiltonian path from a to b in G if and only if there exists a simple path of length k from a to b in G. Hence the reduction from HAMILTONIAN PATH produces G and $k = |V| - 1$. ■

Problem 4 (20 points) Recall the Legendre symbol $(a \,|\, p)$, where p is an odd prime,

$$(a \,|\, p) = \begin{cases} 0, & \text{if } p \,|\, a, \\ 1, & \text{if } a \text{ is a quadratic residue module } p, \\ -1, & \text{if } a \text{ is a quadratic nonresidue module } p. \end{cases}$$

Prove that $\sum\limits_{x=1}^{p} (x \,|\, p) = 0$.

Proof: For a prime p, there exists a primitive root r module p. Obviously, $(r \,|\, p) = -1$. Since $(r, p) = 1$, the map $x \to rx(mod p)$ defines a bijection on the set of residues modulo p. Now,

$$\begin{aligned}\sum_{x=1}^{p}(x \,|\, p) &= \sum_{x=1}^{p}(rx \,|\, p) \\ &= \sum_{x=1}^{p}(r \,|\, p)(x \,|\, p) \\ &= -\sum_{x=1}^{p}(x \,|\, p).\end{aligned}$$

Hence $\sum_{x=1}^{p}(x \,|\, p) = 0$. ■

Problem 5 (20 points) The problem COMPOSITENESS asks if an positive integer N is a composite number. The problem PRIMES asks if an positive integer N is a prime number. We know if N is an odd composite, then $(M \,|\, N) \equiv M^{(N-1)/2} \mod N$ for at most half of $M \in \Phi(N) = \{\, m \,|\, 1 \leq m < N \,,\, \gcd(m, N) = 1 \,\}$.

(1) Decribe a Morte Carlo (randomized) algorithm for COMPOSITENESS and give a brief analysis of the algorithm's error probabilities.

(2) Why is the algorithm not an algorithm for PRIMES?

Ans:

(1) See pp. 586–588 of the lecture notes.

(2) Because it contains false positives (for PRIMES).

■