Graphs and Algorithms

Problem set #1
Due date: 10 Khordad 99 (via lms.iut.ac.ir)

Problem 1. Let P_3 be the path on 3 nodes. We say that an undirected graph $G = (V, E)$ contains an induced P_3 if there exist $u, v, w \in V$ such that $(u, v), (v, w) \in E$ but $(u, w) \notin E$. Design an $O(m + n)$ time algorithm that finds an induced P_3 in a graph G with m-edge and n-node or determines that G does not contain an induced P_3.

Problem 2. Recall the definition of **NP-completeness** that was presented in the class as follows:

A is **NP-complete** if (1) $A \in \mathbf{NP}$ and (2) for all $A' \in \mathbf{NP}$ there exist a function f and a polynomial p such that $f(x)$ is computable in time $p(|x|)$ and $x \in A'$ iff $f(x) \in A$.

Now we give a new definition here as follows.

"A problem like A is para-**NP-complete** if (1) $A \in \mathbf{NP}$ and (2) there exists a polynomial p such that $\forall A' \in \mathbf{NP}$ there exist a function g where $g(x)$ is computable in time $p(|x|)$ and $x \in A'$ if and only if $g(x) \in A$."

Prove that para-**NP-complete** languages do not exist.

Problem 3. For a language L, we define the *Kleene star* of the language is

$$L^* = \{x_1 \dots x_k | x_i \in L, k \geq 0\}.$$

we say that for a complexity class $\mathcal{C}$ closed under Kleene star if $L \in \mathcal{C} \Rightarrow L^* \in \mathcal{C}$.

Show that **P** and **NP** are closed under Kleene star.

Problem 4. In the RATIO-PARTITION problem we are given a set of integers $A = \{a_1, a_2, \dots, a_n\}$, their weights $W = \{w_1, w_2, \dots, w_n\}$ and the number k. We ask whether there exist a subset X of $N = \{1, \dots, n\}$ such that:

$$(\textstyle\sum_{i \in X} w_i \cdot a_i) \bmod k = (\textstyle\sum_{i \in N \setminus X} w_i \cdot a_i) \bmod k.$$

Show that the problem is **NP-complete**.

Problem 5. In the MAX CUT problem, we are given an undirected graph G and an integer K and have to decide whether there is a subset of vertices S such that there are at least K edges that have one endpoint in S and one endpoint in $\overline{S}$. Prove that this problem is NP-complete.

Problem 6. Given a multiset of integers S, in the EQU-SUM problem we want to check whether or not S can be divided into two disjoint subset, say X_1, X_2 such that $\sum_{x_i \in X_1} x_i = \sum_{x_i \in X_2} x_i$. In the 3-EQU-SUM problem for given multiset S we want to check whether S can be divided into three disjoint subset say Y_1, Y_2, Y_3 such that $\sum_{x_i \in Y_1} x_i = \sum_{x_i \in Y_2} x_i = \sum_{x_i \in Y_3} x_i$. Give a direct polynomial reduction from 3-Equ-Sum to Equ-Sum problem.

Problem 7. Suppose that you are given a graph G and a number K and are told that either (i) the minimum vertex cover of G is of size at most K or (ii) it is of size at least $3K$. Show a polynomial-time algorithm that can distinguish between these two cases. Can you do it with a smaller constant than 3? Since VERTEX COVER is NP-hard, why does this algorithm not show that P = NP?

Hint: Use the relationships between the minimum vertex cover and the maximum matching.

Good Luck.