Problem Set 11 for CS 170

Problem 1 [Monotone Satisfiability]

Consider an instance of SAT, specified by clauses $C_1, C_2, \ldots, C_k$ over a set of Boolean variables $x_1, x_2, \ldots, x_n$. We say that the instance is *monotone* if each term in each clause consists of a nonnegated variable; that is, each term is equal to x_i, for some i, rather than $\bar{x}_i$. Monotone instances of SAT are very easy to solve: They are always satisfiable, by setting each variable equal to 1. For example, suppose that we have the three clauses $(x_1 \vee x_2)$, $(x_1 \vee x_3)$, and $(x_2 \vee x_3)$. This is monotone, and indeed the assignment that sets all three variables to 1 satisfies all the clauses. But we can observe that this is not the only satisfying assignment; we could also have set x_1 and x_2 to 1, and x_3 to 0. Indeed, for any monotone instance, it is natural to ask how few variables we need to set to 1 in order to satisfy it.

Given a monotone instance of SAT, together with a number k, the problem of MONOTONE SATISFIABILITY WITH FEW TRUE VARIABLES asks: Is there a satisfying assignment for the instance in which at most k variables are set to 1? Prove that this problem is NP-complete.

Problem 2 [Path Selection]

Consider a communications network modeled by a directed graph $G = (V, E)$. There are c *users* who are interested in making use of this network. User i issues a *request* to reserve a specific path P_i in G on which to transmit data. You are interested in accepting as many of these path requests as possible, subject to the following restriction: if you accept both P_i and P_j, then P_i and P_j cannot share any nodes.

Thus, the PATH SELECTION PROBLEM asks: given a graph G, a set of requests $P_1, \ldots, P_c$ and a number k, is it possible to select at least k of the paths so that no two of the selected paths share any nodes? Prove that PATH SELECTION PROBLEM is NP-complete.

Problem 3 [Attacking Coalitions]

Consider the design of logging software for detecting attacks on a server. Suppose that the software records the IP addresses that users access on the server. Suppose that each user accesses at most one IP address in any given minute; the software writes a log file that records, for each user u and each minute u, a value $I(u, m)$ that is equal to the IP address (if any) accessed by user u during minute m. (It writes a null symbol if there is no such access).

Yesterday the system was attacked. The attack was carried out by accessing i distinct IP addresses over t consecutive minutes: In minute 1, the attack accessed address i_1; in minute 2, the attack accessed address i_2; and so on, to address i_t in minute t.

Checking the logs, it turns out that there is no single user u who accessed each of the IP addresses involved at the appropriate time; in other words, there's no u so that $I(u, m) = i_m$ for each minute m from 1 to t. So the question becomes: what if there were a small *coalition* of k users that collectively carried out the attack? We will say that a subset S of users is a *suspicious coalition* if, for each minute m from 1 to t, there is at least one user $u \in S$ for which $I(u, m) = i_m$. The SUSPICIOUS COALITION PROBLEM asks: Given the collection of all values $I(u, m)$, and a number k, is there a suspicious coalition of size at most k? Prove that this problem is NP-complete.