# CS 473: Fundamental Algorithms, Fall 2011

**Homework 10 (due Tuesday, 23:55:00, November 29, 2011)**

**Collaboration Policy & submission guidelines:** See homework 1.
Each student individually have to also do **quiz 11** online.

Version: **1.00002**

1. (30 PTS.) Beware of Greeks bearing gifts

   (The expression "beware of Greeks bearing gifts" is Based on Virgil's Aeneid: "Quidquid id est, timeo Danaos et dona ferentes", which means literally "Whatever it is, I fear Greeks even when they bring gifts.".)

   The reduction faun, the adopted brother of the Partition satyr, came to visit you on labor day, and left you with four gifts. One of them was 1,000 euros in Greek government bonds – now completely worthless. The other three gifts are black boxes that can solve decision problems. Show how to use these black boxes:

   (A) (10 PTS.) The first black box can solve the following decision problem in polynomial time:

   **Problem:** Minimum Test Collection

   > *Instance:* A finite set $A$ of "possible diagnoses," a collection $C$ of subsets of $A$, representing binary "tests," and a positive integer $J \leq |C|$.
   > *Question:* Is there a subcollection $C' \subseteq C$ with $|C'| \leq J$ such that, for every pair $a_i, a_j$ of possible diagnoses from $A$, there is some test $c \in C'$ for which $|\{a_i, a_j\} \cap c| = 1$ (that is, a test $c$ that "distinguishes" between $a_i$ and $a_j$)?

   Show how to use this black box to solve, in polynomial time, the optimization version of this problem (i.e., finding and outputting the smallest possible set $C'$ that has the desired property).

   (B) (10 PTS.) Next, was a black box for solving Subgraph Isomorphism.

   **Problem:** Subgraph Isomorphism

   > *Instance:* Two graphs, $G = (V_1, E_1)$ and $H = (V_2, E_2)$.
   > *Question:* Does $G$ contain a subgraph *isomorphic* to $H$, that is, a subset $V \subseteq V_1$ and a subset $E \subseteq E_1$ such that $|V| = |V_2|$, $|e| = |E_2|$, and there exists a one-to-one function $f : V_2 \to V$ satisfying $\{u, v\} \in E_2$ if and only if $\{f(u), f(v)\} \in E$?

   Show how to use this black box to compute the subgraph isomorphism (i.e., you are given $G$ and $H$, and you have to output $f$) in polynomial time.

   (C) (10 PTS.) The final black-box can solve Partition in polynomial time (note that this black box solves the decision problem). Let $S$ be a given set of $n$ integer numbers. Describe a polynomial time algorithm that computes, using the black box, a partition of $S$ if such a solution exists. Namely, your algorithm should output a subset $T \subseteq S$, such that

$$\sum_{s \in T} s = \sum_{s \in S \setminus T} s.$$

2. (30 PTS.) The world in base 3.

You are given an arithmetic formula $F$. This formula might use the constants $0, 1, 2$, additions, divisions, multiplications and subtractions (you can also use parenthesis). Naturally, the formula also contains free variables. To make things interesting, all the calculations in this formula are done modulo 3. As such, if $F \equiv x + y$, then for $x = 2$ and $y = 2$, the formula $F$ evaluates to $(2 + 2) \mod 3 = 1$.

In the Equation3 problem, given such formula, you have to decide whether or not there exists an assignment to the free variables of the formula such that it evaluates to 1.

Prove that this problem is NP-COMPLETE by showing a polynomial time reduction from 3SAT to Equation3. (Don't forget also to show that Equation3 is in NP.)

3. (40 PTS.) The strange strange world of MAX 2SAT.

(A) (2 PTS.) Consider two boolean variables $x$ and $y$. Write a 2CNF formula that computes the function $\neg(x \wedge y)$.

(B) (17 PTS.) Given a connected graph $G = (V, E)$ with $n$ vertices and $m$ edges, we want to compute its maximum size independent set. To this end, we define a boolean variable for every vertex of $V$. Describe how to write a 2CNF formula that is true if and only if the vertices that are assigned value 1 are all independent.

(C) (10 PTS.) Describe how to compute a 2CNF formula from $G$ such that there is an assignment that satisfies at least $\Delta$ clauses of this formula if and only if there is an independent set in $G$ of size $k$ (or larger). To make things easy, you are allowed to duplicate the same clause in your formula as many times as you want. Naturally, the algorithm for computing this formula from $G$ should work in polynomial time (and of course, you need to describe this algorithm). (The final formula size has to be polynomial in $n$ and $m$.) What is the value of $\Delta$ as a function of $n$ and $m$?

(D) (10 PTS.) Using the above, prove that MAX 2SAT (i.e., given a 2CNF formula, compute the assignment that maximizes the number of clauses that are satisfied) is an NP-HARD problem. That is, show that if one can solve MAX 2SAT in polynomial time then one can solve 3SAT in polynomial time.

(E) (1 PTS.) As you know, one can solve 2SAT in linear time. Why does this does not imply that MAX 2SAT can be solved in polynomial time?