due Wednesday, 3 December 2014, at 5:00 PM

[Total points: 40]

1. *4 points*

Understanding the Floyd-Warshall algorithm

Do Exercise 25.2-7 on page 700 (page 635 in 2/e). To simplify the notation please omit the implicit and unnecessary (k) superscripts in the Floyd-Warshall algorithm, i.e. use the one described in Exercise 25.2-4, which is equivalent to the one presented in class. And describe your algorithm in terms of computing ϕ_{ij} rather than $\phi_{ij}^{(k)}$.

2. *14 points: 8 for part (a), 3 each for (b) and (c)*

Understanding NP-completeness proofs and conditions under which polynomial-time algorithms exist

The *min-ones satisfiability problem (M1-Sat)* is defined as follows.

Given a formula F in conjunctive normal form and an integer k, does there exist a satisfying assignment A for F such that the number of true variables in A is $\leq k$?

a. Prove that M1-Sat is NP-complete. Be sure to include and clearly mark all parts of the proof, even those that may seem trivial.

b. Let p be the number of variables that have at least one *positive* occurrence – a variable x has a positive occurrence if some clause contains x as a literal (as opposed to $\overline{x}$). Give a $\mathrm{O}(|F|)$ algorithm for solving M1-Sat when p is constant, where $|F|$ is the total number of literals in F, the usual measure of the size of a formula.

c. Determine the fastest growth rate $f(n)$ for which you can come up with a polynomial time algorithm for M1-Sat if $p \in \mathrm{O}(f(n))$. Describe the algorithm and explain why an even slightly faster growth rate would not work (for your algorithm – whether or not a faster growth rate works in general depends on whether P = NP).

3. *12 points: 2 points for (a), 8 for (b), and 2 for (c)*

Purpose: understanding NP*-completeness and reductions among the four variants of a problem: optimization, evaluation, certificate, and decision.*

Do parts (a)-(c) of Problem 34-3 on page 1103 (page 1019 in 2/e) but with the following twist: define the *graph coloring problem* as one whose input is a graph G and whose output is an actual function $c : V \to \{1, \ldots, k\}$ that minimizes k. In other words, the output has to *give the actual color of each vertex* in the k-coloring.

The most important part of this problem is part (b) and, as I pointed out in class, the hardest reduction is from the decision version to the certificate. Here it is important to note that the actual color of a vertex does not matter – what matters is which vertices have the same color.

4. *10 points: 2 points for (a), 8 for (b)*

Understanding NP-completeness proofs

Do Problem 34-4, parts (a) and (b) on page 1104. The problem statement is identical to that of Problem 3 of Homework 3, but there is no restriction on the t_j's. You already did parts (c) and (d) in Homework 3.