CPSC 420+500 Problem Set 6 18 Mar 2017
W. Evans Due: 27 Mar 2017 at 21.00 (9pm)

Submission Instructions

Handin your solutions using `handin`. You can write your solutions by hand and scan the pages or take pictures of them with your phone; or use a word processing package to typeset your solutions. Whatever you do, you should produce **pdf** files **assign6Q1.pdf**, **assign6Q2.pdf**, etc. containing your solutions for Question 1, Question 2, etc.

To handin: Copy your solution files to the directory `~/cs420/a6` in your home directory on a CS undergraduate machine. (You may have to create this directory using `mkdir ~/cs420/a6`.) Then run `handin cs420 a6` from your home directory.

If you do not have a CS department account, you can email your assign1.pdf to me `will@cs.ubc.ca`.

Late submissions are not accepted.

Grading Policy

We will grade a subset of these questions of size at least two. It's a good idea to do all of the questions because (1) you don't know which ones we'll grade and (2) answering these questions is good practice for the exams. We will email feedback to the email address associated with the account you used to handin the assignment.

Questions

Try to answer these on your own but if you work with someone or use an outside source you must acknowledge them in your write-up. Do not copy solutions from any source.

1. (from Problem 8.4 Algorithms by Dasgupta, Papadimitriou, Vazirani) Consider the CLIQUE problem restricted to graphs in which every vertex has degree at most 3. Call this problem CLIQUE-3.

   (a) Prove that CLIQUE-3 is in NP.

   (b) What is wrong with the following proof of NP-completeness for CLIQUE-3?

   We know that the CLIQUE problem in general graphs is NP-complete, so it is enough to present a reduction from CLIQUE-3 to CLIQUE. Given a graph G with vertices of degree ≤ 3, and a parameter k, the reduction leaves the graph and the parameter unchanged. Clearly this is a possible input for the CLIQUE problem. Furthermore, the answer to both problems is identical. This proves the correctness of the reduction and, therefore, the NP-completeness of CLIQUE-3.

   (c) It is true that the VERTEX COVER problem remains NP-complete even when restricted to graphs in which every vertex has degree at most 3. Call this problem VC-3. What is wrong with the following proof of NP-completeness for CLIQUE-3?

   We present a reduction from VC-3 to CLIQUE-3. Given a graph $G = (V, E)$ with node degrees bounded by 3, and a parameter b, we create an instance of CLIQUE-3 by leaving the graph unchanged and switching the parameter to $|V| - b$. Now, a subset $C \subseteq V$ is a vertex cover in G if and only if the complementary set $V - C$ is a clique in G.

Therefore G has a vertex cover of size $\leq b$ if and only if it has a clique of size $\geq |V| - b$. This proves the correctness of the reduction and, consequently, the NP-completeness of CLIQUE-3.

(d) Describe an $O(|V|^4)$-time algorithm for CLIQUE-3.

2. (from Exercise 3 Jeff Erickson's NP-hardness notes) A boolean formula is in *disjunctive normal form* (or DNF) if it consists of a disjunction (Or) of several terms, each of which is the conjunction (And) of one or more literals. For example, the formula

$$(\overline{x} \wedge y \wedge \overline{z}) \vee (y \wedge z) \vee (x \wedge \overline{y} \wedge \overline{z})$$

is in disjunctive normal form. DNF-SAT asks, given a boolean formula in disjunctive normal form, whether that formula is satisfiable.

(a) Describe a polynomial-time algorithm to solve DNF-SAT.

(b) What is the error in the following argument that P=NP?

Suppose we are given a boolean formula in conjunctive normal form with at most three literals per clause, and we want to know if it is satisfiable. We can use the distributive law to construct an equivalent formula in disjunctive normal form. For example,

$$(x \vee y \vee \overline{z}) \wedge (\overline{x} \vee \overline{y}) \Longleftrightarrow (x \wedge \overline{y}) \vee (y \wedge \overline{x}) \vee (\overline{z} \wedge \overline{x}) \vee (\overline{z} \wedge \overline{y})$$

Now we can use the algorithm from part (a) to determine, in polynomial time, whether the resulting DNF formula is satisfiable. We have just solved 3SAT *in polynomial time. Since* 3SAT *is NP-hard, we must conclude that P=NP!*

3. (from Problem 8.14 Algorithms by Dasgupta, Papadimitriou, Vazirani) Prove that the following problem is NP-complete: Given an undirected graph $G = (V, E)$ and an integer k, does G contain a clique of size k *as well as* an independent set of size k?

4. The List Scheduling problem is: Given a list of job times $p_1, p_2, \ldots, p_n$, a number of machines m, and a deadline d, is there a schedule that assigns each job to one of the m identical machines so that all the jobs are completed before the deadline? Job i must run on its assigned machine for p_i time units without interruption. Also a single machine can only work on one job at any time. Prove that List Scheduling is NP-hard.

   You should use the fact that the Partition problem is NP-hard. The Partition problem is: Given a list of integers $x_1, x_2, \ldots, x_n$, is there a subset S of $\{1, 2, \ldots, n\}$ such that $\sum_{i \in S} x_i = \sum_{i \notin S} x_i$?

5. (from Exercise 20 Jeff Erickson's NP-hardness notes) A *tonian path* in a graph G is a path that visits at least half of the vertices of G without visiting any vertex twice. Show that determining if a graph has a tonian path is NP-complete.

   You should use the fact that the HamiltonianPath problem is NP-hard. The HamiltonianPath problem is: Given a graph G, is there a path in G that visits every vertex exactly once?