

**Due:** By 2pm on Monday 26 November.

**Worth:** 10%

This assignment is to be completed in groups of no more than three students. Hand in a single paper for your group, with the information about each student filled in on the cover page.

GENERAL ADVICE: I know many people split up the work on group assignments so that each person is “responsible” for a few questions only. But this assignment is complicated enough that I believe you would greatly benefit from working on it together with your partner(s) rather than trying to solve it individually. Remember that this is supposed to be the point of these group assignments: to give you a chance to work on problems together, so that you each learn more than by doing it yourself.

Also, splitting up the work may save time in the short term, but not in the long term: since everyone is expected to understand how to solve each question, you will each have to go back and review each solution anyway. More importantly, as you well know, there is a big difference between reading someone else’s solution, and working out a solution by yourself: you learn much more by “solving” than by “reading”. Just keep it in mind...

1. [10 marks]

Prove that  $A_{TM}$  is *NP*-hard. Give an explicit  $\mathcal{O}$ -bound on the running time of your reduction function.

2. [10 marks]

We have a network of stations, each one of which needs access to a common database in order to carry out its work (for example, a network of banking machines accessing client information—to simplify, assume all operations are queries, *i.e.*, no operation changes the data). Our problem is to determine which station(s) should store the database (these stations will be called “servers”) in order to have reasonably fast access time (including the load on each server), as well as having as little duplication of the database as possible.

At one extreme, each station could store its own copy of the database. This would ensure the fastest possible access time, but at the cost of replicating the database many times—if the network is large, this is a significant cost. At the other extreme, the database could be stored in a single server (all other stations would connect to the server to access the data). This would ensure the smallest amount of duplication but at the cost of longer access times and server load—if the network is large, this is a significant cost. A reasonable middle ground would be to store the database on some number  $k$  of servers, chosen so that there are few servers and every station is close to a server. Unfortunately, it is impossible to solve this problem efficiently.

Formally, show that the following language is *NP*-complete (where the “distance” between two vertices  $s$  and  $t$  is simply the number of edges on a shortest path from  $s$  to  $t$ , or  $\infty$  if there is no path from  $s$  to  $t$ ).

SL (for “Server Location”) =  $\{ \langle G, d, k \rangle : G \text{ is an undirected graph (the network of stations) for which there is some set } S \text{ of } k \text{ vertices (the servers) such that every vertex in } G \text{ is within distance } d \text{ of some vertex in } S \}$

HINT: Think about special cases for small values of  $d$ ; some of those are close to languages you already know (but not identical—make sure you understand the differences; they are subtle but important).

3. [15 marks]

By analogy with the definition of *NP*-completeness, we say that a language  $A$  is “*coNP*-complete” if

- $A \in \text{coNP}$ , and
- $A$  is *coNP*-hard, *i.e.*,  $\forall B \in \text{coNP}, B \leq_p A$ .

Show that the language EQ defined below is *coNP*-complete.

EQ =  $\{ \langle F_1, F_2 \rangle : F_1 \text{ and } F_2 \text{ are equivalent propositional formulas, } i.e., \text{ each assignment of truth-values to the variables of } F_1 \text{ and } F_2 \text{ gives the same value to both } F_1 \text{ and } F_2 \}$

HINT: First, prove an appropriate relationship between *coNP*-completeness and *NP*-completeness.

4. [10 marks]

Show that the following “Minimum Server Location” optimization problem is self-reducible.

$\text{MSL}(G, d)$ : Given an undirected graph  $G$  and a positive integer  $d$ , return a subset of vertices  $S$  such that every vertex in  $G$  is within distance  $d$  of some vertex in  $S$ , and the size of  $S$  is minimal (*i.e.*, there is no smaller subset  $S'$  with the same property).

5. [15 marks]

The length of a propositional formula  $F$ , denoted  $|F|$ , is the total number of symbols in  $F$ , including each variable, connective, and parenthesis, *e.g.*,  $|(p \wedge q) \rightarrow \neg r| = 8$ . A propositional formula  $F$  is “optimal” if there is no shorter equivalent formula, *i.e.*, for all formulas  $F'$  equivalent to  $F$ ,  $|F'| \geq |F|$ . Consider the following language.

$\text{OPT} = \{ \langle F \rangle : F \text{ is an optimal propositional formula} \}$

(a) [8 marks]

Prove that  $\text{OPT} \in \text{PSPACE}$ .

(b) [7 marks]

Prove that if  $P = \text{NP}$ , then  $\text{OPT} \in P$ .

HINT: Recall that  $P = \text{coP}$ , and have another look at the rest of this assignment...