Test 2
(Sample solution)

First name: ________________________ Student number: |_|_|_|_|_|_|_|_|_|

Last name: ________________________ TA's name: ________________________

Tutorial room: ________________________

Read the following **carefully** before starting.

1. There are six pages to this test, including this one.
2. The duration of the test is 50 minutes. No aids are allowed.
3. The last two pages contain a list of NP-complete problems. In your answers, you may assume that these problems, and **only** these problems, are NP-complete.
4. Write your student number at the bottom of each of the first four pages.
5. All answers must be written on this test booklet. You may use the back of the page for extra space. Scrap paper will be provided, but will not be marked.
6. If you state clearly that you do not know the answer to a question, you will receive 20% of the marks for that question.

Part I	______/6
Part II	______/10
Part III	______/14
Total	______/30

Part I (6 marks)

1. Suppose that A and B are languages, and that $A \leq_p B$. Circle one answer for each statement below.

If $A \in \mathbf{P}$ then $B \in \mathbf{P}$.

TRUE $\boxed{\text{FALSE}}$ I DON'T KNOW

If $B \in \mathbf{P}$ then $A \in \mathbf{P}$.

$\boxed{\text{TRUE}}$ FALSE I DON'T KNOW

If A is **NP**-hard then B is **NP**-hard.

$\boxed{\text{TRUE}}$ FALSE I DON'T KNOW

If B is **NP**-hard then A is **NP**-hard.

TRUE $\boxed{\text{FALSE}}$ I DON'T KNOW

2. Write down the formal definition of $A \leq_p B$.

Solution: There exists a function $f : \Sigma^* \mapsto \Sigma^*$ that is computable in polynomial time, such that for all $x \in \Sigma^*$:

$$x \in A \iff f(x) \in B$$

Student number: __________

Part II (10 marks)

A simple path in a graph is a path that does not contain any repeated vertices. The length of a path is the number of edges in the path. The decision problem LONGEST-PATH is defined below.

Instance: A directed graph $G = (V, E)$, and $k \in \mathbb{N}$
Question: Does G have a simple path of length k or longer?

Prove that LONGEST-PATH is **NP**-hard (note: you do not need to prove that it is in **NP**). Your answer will be marked on its structure as well as its content.

Solution: Show HAM-PATH-EXISTS $\leq_p$ LONGEST-PATH. Note that a Hamiltonian path is a simple path of length $n - 1$, and conversely a simple path of length $n - 1$ is a Hamiltonian path. Define $f(\langle G\rangle) = \langle G, n - 1\rangle$, where n is the number of vertices in G. Clearly f is computable in polynomial time: an algorithm for computing f simply counts the number n of vertices in the input graph and appends the number $n - 1$ to its input.

If $\langle G\rangle \in$ HAM-PATH-EXISTS, then G has a Hamiltonian path, i.e. G has a simple path of length $n - 1$, so $f(\langle G\rangle) = \langle G, n - 1\rangle \in$ LONGEST-PATH.

Conversely, if $f(\langle G\rangle) = \langle G, n-1\rangle \in$ LONGEST-PATH, then G has a simple path with $n-1$ edges, and thus with n vertices. As a simple path with n vertices is a Hamiltonian path, it follows that $\langle G\rangle \in$ HAM-PATH-EXISTS.

We have shown that f is computable in polynomial time, and for all graphs G, $\langle G\rangle \in$ HAM-PATH-EXISTS $\iff f(\langle G\rangle) \in$ LONGEST-PATH. Thus HAM-PATH-EXISTS $\leq_p$ LONGEST-PATH. Since HAM-PATH-EXISTS is **NP**-hard it follows that LONGEST-PATH is **NP**-hard.

Student number: ____________

Part III (14 marks)

A language is **co-NP**-complete if its complement is **NP**-complete.

A boolean formula φ is a tautology if $\varphi(\tau) = 1$ for every truth assignment τ. The decision problem TAUTOLOGY is:

Instance: A boolean formula φ
Question: Is φ a tautology?

Prove that TAUTOLOGY is **co-NP**-complete. Your answer will be marked on its structure as well as its content.

Solution: The complement of TAUTOLOGY is the language FALSIFIABLE, consisting of all boolean formulae φ for which there exists a falsifying assignment τ.

Instance: A boolean formula φ
Question: Does there exist a truth assignment τ such that $\varphi(\tau) = 0$?

To prove that TAUTOLOGY is **co-NP**-complete, we must prove that FALSIFIABLE is **NP**-complete, i.e. we must prove that it is in **NP**, and that it is **NP**-hard. Clearly FALSIFIABLE is in **NP**: the non-deterministic algorithm guesses a truth assignment τ and checks that $\varphi(\tau) = 0$. As evaluating a formula can be done in polynomial time, this algorithm runs in (non-deterministic) polynomial time.

To prove that FALSIFIABLE is **NP**-hard, we show SAT $\leq_p$ FALSIFIABLE. Note that a formula φ is satisfiable iff its negation $\neg(\varphi)$ is falsifiable. The reduction is therefore defined as

$$f(\langle\varphi\rangle) = \langle\neg(\varphi)\rangle$$

Clearly f can be computed in polynomial time: an algorithm for computing f simply encloses its input in brackets and prepends a negation symbol $\neg$.

If $\langle\varphi\rangle \in$ SAT then there exists a truth assignment τ such that $\varphi(\tau) = 1$, and thus $\neg(\varphi)(\tau) = 0$, implying $\langle\neg(\varphi)\rangle = f(\langle\varphi\rangle) \in$ FALSIFIABLE.

Conversely, if $\langle\neg(\varphi)\rangle = f(\langle\varphi\rangle) \in$ FALSIFIABLE, then there exists a truth assignment τ such that $\neg(\varphi)(\tau) = 0$, and thus $\varphi(\tau) = 1$, implying $\langle\varphi\rangle \in$ SAT.

We have shown that $\langle\varphi\rangle \in \text{SAT} \iff f(\langle\varphi\rangle) \in \text{FALSIFIABLE}$, and f is computable in polynomial time, so SAT $\leq_p$ FALSIFIABLE. Since SAT is **NP**-hard it follows that FALSIFIABLE is **NP**-hard.

Student number: _ _ _ _ _ _ _ _ _

Appendix: NP-complete problems

SAT

Instance: A boolean formula φ
Question: Is φ satisfiable?

3SAT

Instance: A boolean formula φ in 3CNF
Question: Is φ satisfiable?

SUBSET-SUM

Instance: $s_1, \ldots, s_n, t \in \mathbb{Z}$
Question: Does there exist a subset $S \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in S} s_i = t$?

PARTITION

Instance: $s_1, \ldots, s_n \in \mathbb{Z}$
Question: Does there exist $S \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in S} s_i = \sum_{j \notin S} s_j$?

KNAPSACK

Instance: $w_1, \ldots, w_n \in \mathbb{N}$; $p_1, \ldots, p_n \in \mathbb{N}$; $W, P \in \mathbb{N}$
Question: Does there exist $S \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in S} w_i \leq W$ and $\sum_{i \in S} p_i \geq P$?

VERTEX-COVER

Instance: A graph $G = (V, E)$, and $k \in \mathbb{N}$
Question: Does G have a vertex cover of size k or smaller?

INDEPENDENT-SET

Instance: A graph $G = (V, E)$ and $k \in \mathbb{N}$
Question: Does G have an independent set of size k or larger?

CLIQUE

Instance: A graph $G = (V, E)$, and $k \in \mathbb{N}$
Question: Does G have a clique of size k or larger?

SET-COVER

Instance: Sets $S_1, \ldots, S_n$, and $k \in \mathbb{N}$
Question: Does there exist $I \subseteq \{1, \ldots, n\}$ such that $|I| \leq k$ and $\bigcup_{i \in I} S_i = \bigcup_{i=1}^{n} S_i$?

HAM-PATH

Instance: A directed graph $G = (V, E)$, and $s, t \in V$
Question: Does G have a Hamiltonian path from s to t?

HAM-CYCLE

Instance: A directed graph $G = (V, E)$
Question: Does G have a Hamiltonian cycle?

HAM-PATH-EXISTS

Instance: A directed graph $G = (V, E)$
Question: Does G have a Hamiltonian path (between any two vertices)?

TSP

Instance: A distance function $d : \{1, \ldots, n\} \times \{1, \ldots, n\} \mapsto \mathbb{N}$, and $k \in \mathbb{N}$
Question: Does there exist a permutation $\varphi : \{1, \ldots, n\} \mapsto \{1, \ldots, n\}$ such that

$$\sum_{i=1}^{i-1} d(\varphi(i), \varphi(i+1)) + d(\varphi(n), \varphi(1)) \leq k$$