NOTE TO STUDENTS: This file contains sample solutions to the term test together with the marking scheme and comments for each question. Please read the solutions and the marking schemes and comments carefully. Make sure that you understand why the solutions given here are correct, that you understand the mistakes that you made (if any), and that you understand *why* your mistakes were mistakes.

Remember that although you may not agree completely with the marking scheme given here it was followed the same way for all students. We will remark your test only if you clearly demonstrate that the marking scheme was not followed correctly. We will make *no* exception to the marking scheme, unless you can clearly demonstrate that it is somehow incorrect.

For all remarking requests, please submit your request **in writing** directly to your instructor. For all other questions, please don't hesitate to ask your instructor during office hours or by e-mail.

Question 1. [8 MARKS]

Part (a) [2 MARKS]

Complete the following definition (check all that are true): "*NP* is defined to be the set of...

- [ ] ...nondeterministic Turing machines that run in polynomial time."
- [ ] ...languages that are not decidable in polynomial time."
- [x] ...all sets of strings decidable in nondeterministic polynomial time."
- [ ] ...problems that can be solved in deterministic polynomial time."

SAMPLE SOLUTION: (See above.)

MARKING SCHEME:

A. .5 marks for each box correctly checked or unchecked

EXPECTED ERROR:

- checking the first box (*NP* is a set of languages, not algorithms)

Question 1. (CONTINUED)

Indicate whether each statement below is true or false (circle the appropriate answer), and briefly justify.

Part (b) [2 MARKS]

For all languages A, if $A \in P$, then $A \in NP$.

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

Justification:

SAMPLE SOLUTION:

$P \subseteq NP$; or
every polytime deterministic algorithm (for deciding a language) is a polytime nondeterministic algorithm

MARKING SCHEME:

A. 1 mark: correct answer ("I DON'T KNOW" worth 0.4)
B. 1 mark: good justification (give mark for correctly justifying an incorrect answer)
"A is no harder to solve than A" deserves 0.5 marks

- Some students may interpret "I DON'T KNOW" to mean "it is unknown", in which case they should get the mark for justification when appropriate. (This applies to each part of this question.)

Part (c) [2 MARKS]

For all languages A, if $SAT \leq_p A$, then A is NP-complete.

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

Justification:

SAMPLE SOLUTION:

This is the definition of NP-hard; for A to be NP-complete it must also be in NP

MARKING SCHEME:

A. 1 mark: correct answer ("I DON'T KNOW" worth 0.4)
B. 1 mark: good justification (give mark for correctly justifying an incorrect answer)

Part (d) [2 MARKS]

For all languages A and B, if $A \leq_p B$ and $B \leq_p A$, then $A \in P$ iff $B \in P$.

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

Justification:

SAMPLE SOLUTION:

If $A \leq_p B$ and $B \in P$, then we know $A \in P$. Same for B.

MARKING SCHEME:

A. 1 mark: correct answer ("I DON'T KNOW" worth 0.4)
B. 1 mark: good justification (give mark for correctly justifying an incorrect answer)

Question 2. [8 MARKS]

For each language below, circle the **smallest** class that the language belongs to and justify your answer.

Part (a) [4 MARKS]

TAUT $= \{\langle F\rangle : F$ is a Boolean formula that is always true (for any truth assignment of its variables)$\}$ belongs to:

P *NP* $\boxed{coNP}$ NONE OF THESE CLASSES I DON'T KNOW

Justification:

SAMPLE SOLUTION:

The complement of the language belongs to *NP*, because it takes polytime to check that a certificate represents a falsifying truth assignment.

MARKING SCHEME:

A. 1 mark: correct answer ("I DON'T KNOW" worth 0.8)
B. 1 mark: correct justification structure (polytime algorithm for *P*, polytime verifier for *NP*, polytime verifier for the complement for *coNP*), whether or not justification itself is correct
C. 2 marks: correct justification (give marks for correctly justifying an incorrect answer, maybe 1/2 for a good effort at justifying answer *P* or *NP*)

Part (b) [4 MARKS]

NOKCLIQUE $= \{\langle G\rangle : G$ is an undirected graph that contains **no** clique with exactly k vertices$\}$ (k is a constant independent of G) belongs to:

$\boxed{P}$ *NP* *coNP* NONE OF THESE CLASSES I DON'T KNOW

Justification:

SAMPLE SOLUTION:

Since k is a constant, there are $\binom{n}{k} \in \Theta(n^k)$ subsets of k vertices, and it takes polytime to check each one and make sure that it does not form a clique (there are only $\Theta(k^2)$ possible edges to check for each subset, a constant).

MARKING SCHEME:

A. 1 mark: correct answer ("I DON'T KNOW" worth 0.8)
B. 1 mark: correct justification structure (polytime algorithm for *P*, polytime verifier for *NP*, polytime verifier for the complement for *coNP*), whether or not justification itself is correct
C. 2 marks: correct justification (give marks for correctly justifying an incorrect answer, except at most 1/2 for correctly justifying answer *NP*)

Question 3. [14 MARKS]

Give a detailed proof that the following language is *NP*-complete. A significant portion of your mark will be based on the structure of your answer, so write your proof carefully. In your answer, you may use any of the known *NP*-complete languages listed at the bottom of the last page.

$$\text{TSP} = \{ \langle G, k \rangle : G \text{ is a complete weighted undirected graph (every edge has a "cost" } c(e)) \text{ that contains a simple cycle visiting every vertex of } G \text{ such that the sum of the weights on the edges of the cycle is at most } k \}$$

A *complete graph* has an edge between each pair of vertices.

SAMPLE SOLUTION:

TSP $\in$ *NP*:
it takes polytime to verify that a certificate encodes a cycle on n vertices (a Hamiltonian cycle) whose total weight is at most k.
TSP is *NP*-hard because HAMCYCLE $\leq_p$ TSP:
On input $\langle G \rangle$, construct $\langle G', k \rangle$ as follows: Construct $G' = (V', E', c)$ from $G = (V, E)$ by setting $V' = V$ and E' = all possible edges. Set c to be such that $c(e) = 1$ if $e \in E$, and $c(e) = n + 1$ if $e \notin E$. (Note that any large number works for this second case.) Finally, set $k = n$ (where $n = |V|$). Clearly, this can be done in polytime.

If G contains a Hamiltonian cycle, then G' will contain a tour of weight n (just use the same cycle: it will only use edges of weight 1).

If G' contains a tour of all n vertices whose total weight is at most n, then it can only ever use edges of weight 1. Thus the tour in G' is also a cycle in G (only the edges of G are used) on n vertices, a Hamiltonian cycle.

MARKING SCHEME:

- In what follows, "explicit/obvious attempt to do X" means either student explicitly states "I will now do X" (whether or not they actually proceed to do it) or student obviously carries out the necessary steps for X (whether or not they state that they are doing it).

A. 1 mark: explicit/obvious attempt to prove TSP $\in$ *NP* and TSP is *NP*-hard

B. 1 mark: explicit/obvious attempt to describe a certificate/verifier for TSP

C. 2 marks: correct certificate/verifier that works in polytime

D. 1 mark: explicit/obvious attempt to show $D \leq_p$ TSP for some *NP*-hard D

E. 3 marks: correct overall format of reduction (describe explicit construction for $\langle G, k \rangle$ given input x for D; argue construction is computable in polytime and $x \in D$ iff $\langle G, k \rangle \in$ TSP), whether or not reduction is correct (simply ensure the required steps are present)

F. 3 marks: correct construction (polytime computable, works)

G. 3 marks: correct argument of construction correctness