"CS 374": Algorithms and Models of Computation, Fall 2014

Final Exam (Version B) — December 16, 2014

Name:	
NetID:	
Section:	1 2 3

#	1	2	3	4	5	6	Total
Score							
Max	20	10	10	10	10	10	70
Grader							

- ***Don't panic!***
- Please print your name and your NetID and circle your discussion section in the boxes above.
- This is a closed-book, closed-notes, closed-electronics exam. If you brought anything except your writing implements and your two double-sided 8½" × 11" cheat sheets, please put it away for the duration of the exam. In particular, you may not use *any* electronic devices.
- **Please read the entire exam before writing anything.** Please ask for clarification if any question is unclear.
- **You have 180 minutes.**
- If you run out of space for an answer, continue on the back of the page, or on the blank pages at the end of this booklet, **but please tell us where to look.** Alternatively, feel free to tear out the blank pages and use them as scratch paper.
- **Please return your cheat sheets and all scratch paper with your answer booklet.**
- If you use a greedy algorithm, you must prove that it is correct to receive credit. **Otherwise, proofs are required only if we specifically ask for them.**
- As usual, answering any (sub)problem with "I don't know" (and nothing else) is worth 25% partial credit. **Yes, even for problem 1.** Correct, complete, but suboptimal solutions are *always* worth more than 25%. A blank answer is not the same as "I don't know".
- ***Good luck!*** And have a great winter break!

---

1. For each of the following questions, indicate ***every*** correct answer by marking the "Yes" box, and indicate ***every*** incorrect answer by marking the "No" box. **Assume P ≠ NP.** If there is any other ambiguity or uncertainty, mark the "No" box. For example:

~~Yes~~ (X)	No	$2+2=4$
Yes	~~No~~ (X)	$x+y=5$
Yes	~~No~~ (X)	3SAT can be solved in polynomial time.
~~Yes~~ (X)	No	Jeff is not the Queen of England.

There are 40 yes/no choices altogether, each worth ½ point.

(a) Which of the following statements is true for ***every*** language $L \subseteq \{0,1\}^*$?

Yes	No	L is non-empty.
Yes	No	L is decidable or L is infinite (or both).
Yes	No	L is accepted by some DFA with 42 states if and only if L is accepted by some NFA with 42 states.
Yes	No	If L is regular, then $L \in$ NP.
Yes	No	L is decidable if and only if its complement $\overline{L}$ is undecidable.

(b) Which of the following computational models can simulate a deterministic Turing machine with three read/write heads, with at most polynomial slow-down in time, assuming P ≠ NP?

Yes	No	A C++ program
Yes	No	A deterministic Turing machine with one head
Yes	No	A deterministic Turing machine with 3 tapes, each with 5 heads
Yes	No	A nondeterministic Turing machine with one head
Yes	No	A nondeterministic finite-state automaton (NFA)

(c) Which of the following languages are decidable?

Yes	No	$\varnothing$
Yes	No	$\{ww \mid w \text{ is a palindrome}\}$
Yes	No	$\{\langle M\rangle \mid M \text{ is a Turing machine}\}$
Yes	No	$\{\langle M\rangle \mid M \text{ accepts } \langle M\rangle \bullet \langle M\rangle\}$
Yes	No	$\{\langle M\rangle \mid M \text{ accepts an infinite number of palindromes}\}$
Yes	No	$\{\langle M\rangle \mid M \text{ accepts } \varnothing\}$
Yes	No	$\{\langle M, w\rangle \mid M \text{ accepts } www\}$
Yes	No	$\{\langle M, w\rangle \mid M \text{ accepts } w \text{ after } \textbf{at least } \lvert w\rvert^2 \text{ transitions}\}$
Yes	No	$\{\langle M, w\rangle \mid M \text{ changes a non-blank on the tape to a blank, given input } w\}$
Yes	No	$\{\langle M, w\rangle \mid M \text{ changes a blank on the tape to a non-blank, given input } w\}$

(d) Let M be a standard Turing machine (with a single one-track tape and a single head) such that ACCEPT(M) is the regular language `0*1*`. Which of the following ***must*** be true?

Yes	No	Given an empty initial tape, M eventually halts.
Yes	No	M accepts the string `1111`.
Yes	No	M rejects the string `0110`.
Yes	No	M moves its head to the right at least once, given input `1100`.
Yes	No	M moves its head to the right at least once, given input `0101`.
Yes	No	M must read a blank before it accepts.
Yes	No	For some input string, M moves its head to the left at least once.
Yes	No	For some input string, M changes at least one symbol on the tape.
Yes	No	M always halts.
Yes	No	If M accepts a string w, it does so after at most $O(\lvert w\rvert^2)$ steps.

(e) Consider the following pair of languages:

- HAMILTONIANPATH := $\{G \mid G$ contains a Hamiltonian path$\}$
- CONNECTED := $\{G \mid G$ is connected$\}$

Which of the following ***must*** be true, assuming P≠NP?

Yes	No	CONNECTED ∈ NP
Yes	No	HAMILTONIANPATH ∈ NP
Yes	No	HAMILTONIANPATH is undecidable.
Yes	No	There is a polynomial-time reduction from HAMILTONIANPATH to CONNECTED.
Yes	No	There is a polynomial-time reduction from CONNECTED to HAMILTONIANPATH.

(f) Suppose we want to prove that the following language is undecidable.

$$\text{ALWAYSHALTS} := \{\langle M\rangle \mid M \text{ halts on every input string}\}$$

Bullwinkle J. Moose suggests a reduction from the standard halting language

$$\text{HALT} := \{\langle M, w\rangle \mid M \text{ halts on inputs } w\}.$$

Specifically, suppose there is a Turing machine AH that decides ALWAYSHALTS. Bullwinkle claims that the following Turing machine H decides HALT. Given an arbitrary encoding $\langle M, w\rangle$ as input, machine H writes the encoding $\langle M'\rangle$ of a new Turing machine M' to the tape and passes it to AH, where M' implements the following algorithm:

```
M'(x):
  if M accepts w
     reject
  if M rejects w
     accept
```

Which of the following statements is true for all inputs $\langle M, w\rangle$?

Yes	No	If M accepts w, then M' halts on every input string.
Yes	No	If M rejects w, then M' halts on every input string.
Yes	No	If M rejects w, then H rejects $\langle M, w\rangle$.
Yes	No	If M diverges on w, then H diverges on $\langle M, w\rangle$.
Yes	No	H does not correctly decide the language HALT. (That is, Bullwinkle's reduction is incorrect.)

---

2. A ***near-Hamiltonian cycle*** in a graph G is a closed walk in G that visits one vertex exactly twice and every other vertex exactly once.

   (a) Give an example of a graph that contains a near-Hamiltonian cycle, but does not contain a Hamiltonian cycle (which visits every vertex exactly once).

   (b) ***Prove*** that it is NP-hard to determine whether a given graph contains a near-Hamiltonian cycle.

3. Give a complete, formal, self-contained description of a DFA that accepts all strings in $\{0,1\}^*$ such that every fifth bit is `0` and the length is *not* divisible by 12. For example, your DFA should accept the strings `11110111101` and `11`. Specifically:

   (a) What are the states of your DFA?

   (b) What is the start state of your DFA?

   (c) What are the accepting states of your DFA?

   (d) What is your DFA's transition function?

4. Suppose you are given three strings $A[1..n]$, $B[1..n]$, and $C[1..n]$. Describe and analyze an algorithm to find the maximum length of a common subsequence of all three strings. For example, given the input strings

$$A = \texttt{AxxBxxCDxEF}, \qquad B = \texttt{yyABCDyEyFy}, \qquad C = \texttt{zAzzBCDzEFz},$$

your algorithm should output the number 6, which is the length of the longest common subsequence `ABCDEF`.

5. For each of the following languages over the alphabet $\Sigma = \{0, 1\}$, either ***prove*** that the language is regular, or ***prove*** that the language is not regular.

   (a) $\{www \mid w \in \Sigma^*\}$

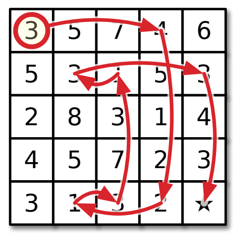
   (b) $\{wxw \mid w, x \in \Sigma^*\}$

6. A ***number maze*** is an $n \times n$ grid of positive integers. A token starts in the upper left corner; your goal is to move the token to the lower-right corner.

   - On each turn, you are allowed to move the token up, down, left, or right.
   - The distance you may move the token is determined by the number on its current square. For example, if the token is on a square labeled 3, then you may move the token three steps up, three steps down, three steps left, or three steps right.
   - However, you are never allowed to move the token off the edge of the board. In particular, if the current number is too large, you may not be able to move at all.

   Describe and analyze an efficient algorithm that either returns the minimum number of moves required to solve a given number maze, or correctly reports that the maze has no solution. For example, given the maze shown below, your algorithm would return the number 8.

3	5	7	4	6
5	3	1	5	3
2	8	3	1	4
4	5	7	2	3
3	1	3	2	★



A 5×5 number maze that can be solved in eight moves.

(scratch paper)

(scratch paper)

You may assume the following problems are NP-hard:

CircuitSat: Given a boolean circuit, are there any input values that make the circuit output True?

3Sat: Given a boolean formula in conjunctive normal form, with exactly three literals per clause, does the formula have a satisfying assignment?

MaxIndependentSet: Given an undirected graph G, what is the size of the largest subset of vertices in G that have no edges among them?

MaxClique: Given an undirected graph G, what is the size of the largest complete subgraph of G?

MinVertexCover: Given an undirected graph G, what is the size of the smallest subset of vertices that touch every edge in G?

3Color: Given an undirected graph G, can its vertices be colored with three colors, so that every edge touches vertices with two different colors?

HamiltonianPath: Given an undirected graph G, is there a path in G that visits every vertex exactly once?

HamiltonianCycle: Given an undirected graph G, is there a cycle in G that visits every vertex exactly once?

DirectedHamiltonianCycle: Given an directed graph G, is there a directed cycle in G that visits every vertex exactly once?

TravelingSalesman: Given a graph G (either directed or undirected) with weighted edges, what is the minimum total weight of any Hamiltonian path/cycle in G?

Draughts: Given an $n \times n$ international draughts configuration, what is the largest number of pieces that can (and therefore must) be captured in a single move?

Super Mario: Given an $n \times n$ level for Super Mario Brothers, can Mario reach the castle?

You may assume the following languages are undecidable:

$$\text{SelfReject} := \left\{ \langle M \rangle \;\middle|\; M \text{ rejects } \langle M \rangle \right\}$$
$$\text{SelfAccept} := \left\{ \langle M \rangle \;\middle|\; M \text{ accepts } \langle M \rangle \right\}$$
$$\text{SelfHalt} := \left\{ \langle M \rangle \;\middle|\; M \text{ halts on } \langle M \rangle \right\}$$
$$\text{SelfDiverge} := \left\{ \langle M \rangle \;\middle|\; M \text{ does not halt on } \langle M \rangle \right\}$$
$$\text{Reject} := \left\{ \langle M, w \rangle \;\middle|\; M \text{ rejects } w \right\}$$
$$\text{Accept} := \left\{ \langle M, w \rangle \;\middle|\; M \text{ accepts } w \right\}$$
$$\text{Halt} := \left\{ \langle M, w \rangle \;\middle|\; M \text{ halts on } w \right\}$$
$$\text{Diverge} := \left\{ \langle M, w \rangle \;\middle|\; M \text{ does not halt on } w \right\}$$
$$\text{NeverReject} := \left\{ \langle M \rangle \;\middle|\; \text{Reject}(M) = \varnothing \right\}$$
$$\text{NeverAccept} := \left\{ \langle M \rangle \;\middle|\; \text{Accept}(M) = \varnothing \right\}$$
$$\text{NeverHalt} := \left\{ \langle M \rangle \;\middle|\; \text{Halt}(M) = \varnothing \right\}$$
$$\text{NeverDiverge} := \left\{ \langle M \rangle \;\middle|\; \text{Diverge}(M) = \varnothing \right\}$$