True or False? (from previous final exams)

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 $\neq$ NP.** If there is any other ambiguity or uncertainty about an answer, mark the "No" box. For example:

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

On an actual exam, each correct choice would be worth ½ point, and each incorrect choice would be worth −¼ point.

(a) Let M be a standard Turing machine (with a single one-track tape and a single head) that ***decides*** 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 never accepts before reading a blank.
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.

(b) Recall the halting language $\textsc{Halt} = \{\langle M, w\rangle \mid M \text{ halts on input } w\}$. Which of the following statements about its complement $\overline{\textsc{Halt}} = \Sigma^* \setminus \textsc{Halt}$ are true?

Yes	No	$\overline{\textsc{Halt}}$ is empty.
Yes	No	$\overline{\textsc{Halt}}$ is regular.
Yes	No	$\overline{\textsc{Halt}}$ is infinite.
Yes	No	$\overline{\textsc{Halt}}$ is decidable.
Yes	No	$\overline{\textsc{Halt}}$ is acceptable but not decidable.
Yes	No	$\overline{\textsc{Halt}}$ is not acceptable.

(c) Suppose some language $A \in \{0, 1\}^*$ reduces to another language $B \in \{0, 1\}^*$. Which of the following statements ***must*** be true?

Yes	No	A Turing machine that recognizes A can be used to construct a Turing machine that recognizes B.
Yes	No	A is decidable.
Yes	No	If B is decidable then A is decidable.
Yes	No	If A is decidable then B is decidable.
Yes	No	If B is NP-hard then A is NP-hard.

(d) Suppose there is a ***polynomial-time*** reduction from problem A to problem B. Which of the following statements ***must*** be true?

Yes	No	Problem B is NP-hard.
Yes	No	A polynomial time algorithm for B can be used to solve A in polynomial time.
Yes	No	If B has no polynomial-time algorithm then neither does A.
Yes	No	If A is NP-hard and B has a polynomial-time algorithm then $\mathrm{P} = \mathrm{NP}$.
Yes	No	If B is NP-hard then A is NP-hard.
Yes	No	If B is undecidable then A is undecidable.

(e) Consider an arbitrary language $L \subseteq \{0, 1\}^*$. Which of the following statements ***must*** be true?

Yes	No	L is non-empty.
Yes	No	L is infinite.
Yes	No	L contains the empty string.
Yes	No	If L is decidable, then L is infinite.
Yes	No	If L is not decidable, then L is infinite.
Yes	No	If L is the union of two regular languages, then its complement $\overline{L}$ is context-free.
Yes	No	If L is finite, then L is context-free.
Yes	No	L is decidable or L is infinite (or both).
Yes	No	If L is the union of two regular languages, then its complement $\overline{L}$ is regular.
Yes	No	L is accepted by some DFA if and only if L is accepted by some NFA.
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 not regular, then L is undecidable.
Yes	No	If L has an infinite fooling set, then L is undecidable.
Yes	No	L is decidable if and only if its complement $\overline{L}$ is undecidable.

(f) Which of the following problems are decidable?

Yes	No	$\varnothing$
Yes	No	$\{\mathtt{0}^n\mathtt{1}^n\mathtt{0}^n\mathtt{1}^n \mid n \geq 0\}$
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	$\{\langle M\rangle \mid M \text{ accepts a finite number of non-palindromes}\}$
Yes	No	$\{\langle M\rangle \mid M \text{ accepts } \varnothing\}$
Yes	No	$\{\langle M, w\rangle \mid M \text{ accepts } w^R\}$
Yes	No	Given an NFA N, is the language $L(N)$ infinite?
Yes	No	Given a context-free grammar G and a string w, is w in the language $L(G)$?
Yes	No	Given an undirected graph G, does G contain a Hamiltonian cycle?
Yes	No	Given two Turing machines M and M', is there a string w that is accepted by both M and M'?

(g) Consider the following pair of languages:

- $\textsc{HamiltonianPath} := \{G \mid G \text{ contains a Hamiltonian path}\}$
- $\textsc{Connected} := \{G \mid G \text{ is connected}\}$

Which of the following ***must*** be true, assuming P$\neq$NP?

Yes	No	$\textsc{Connected} \in \text{NP}$
Yes	No	$\textsc{HamiltonianPath} \in \text{NP}$
Yes	No	$\textsc{HamiltonianPath}$ is decidable.
Yes	No	There is no polynomial-time reduction from $\textsc{HamiltonianPath}$ to $\textsc{Connected}$.
Yes	No	There is no polynomial-time reduction from $\textsc{Connected}$ to $\textsc{HamiltonianPath}$.

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

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

Rocket J. Squirrel a reduction from the standard halting language

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

Specifically, given a Turing machine DECIDEALWAYSHALTS that decides ALWAYSHALTS, Rocky claims that the following Turing machine DECIDEHALT decides HALT.

```
DecideHalt(⟨M, w⟩):
  Encode the following Turing machine M′:
      M′(x):
        if M accepts w
            reject
        if M rejects w
            accept
  return DecideAlwaysHalts(⟨M′⟩)
```

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 diverges on w, then M' halts on every input string.
Yes	No	If M accepts w, then DECIDEALWAYSHALTS accepts $\langle M'\rangle$.
Yes	No	If M rejects w, then DECIDEHALT rejects $\langle M, w\rangle$.
Yes	No	If M diverges on w, then DECIDEALWAYSHALTS diverges on $\langle M'\rangle$.
Yes	No	DECIDEHALT decides HALT. (That is, Rocky's reduction is correct.)

NP-hardness

2. A ***relaxed 3-coloring*** of a graph G assigns each vertex of G one of three colors (for example, red, green, and blue), such that ***at most one*** edge in G has both endpoints the same color.

   (a) Give an example of a graph that has a relaxed 3-coloring, but does not have a proper 3-coloring (where every edge has endpoints of different colors).

   (b) ***Prove*** that it is NP-hard to determine whether a given graph has a relaxed 3-coloring.

3. An ***ultra-Hamiltonian cycle*** in G is a closed walk C that visits every vertex of G exactly once, except for *at most one* vertex that C visits more than once.

   (a) Give an example of a graph that contains a ultra-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 ultra-Hamiltonian cycle.

4. An ***infra-Hamiltonian cycle*** in G is a closed walk C that visits every vertex of G exactly once, except for *at most one* vertex that C does not visit at all.

   (a) Give an example of a graph that contains a infra-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 infra-Hamiltonian cycle.

5. A ***quasi-satisfying assignment*** for a 3CNF boolean formula Φ is an assignment of truth values to the variables such that *at most one* clause in Φ does not contain a true literal. ***Prove*** that it is NP-hard to determine whether a given 3CNF boolean formula has a quasi-satisfying assignment.

6. Jerry Springer and Maury Povich have decided not to compete with each other over scheduling guests during the next talk-show season. There is only one set of Weird People who either host would consider having on their show. The hosts want to divide the Weird People into two (disjoint) groups: those to appear on Jerry's show, and those to appear on Maury's show. (Neither wants to "recycle" a guest that appeared on the other's show.)

   Both Jerry and Maury have preferences about which Weird People they are particularly interested in. For example, Jerry wants to be sure to get at least one person who fits the category "had extra-terrestrial affair". Thus, on his list of preferences, he writes "w_1 or w_3 or w_{45}", since weird people numbered 1, 3, and 45 are the only ones who fit that description. Jerry has other preferences as well, so he lists those also. Similarly, Maury might like to guarantee that his show includes at least one guest who confesses to "really enjoying Rice's theorem". Each potential guest may fall into any number of different categories, such as the person who enjoys Rice's theorem more than the extra-terrestrial affair they had.

   Jerry and Maury each prepare a list reflecting all of their preferences. Each list contains a collection of statements of the form "(w_i or w_j or w_k)". Your task is to prove that it is NP-hard to find an assignment of weird guests to the two shows that satisfies all of Jerry's preferences and all of Maury's preferences.

(a) The problem NoMixedClauses3Sat is the special case of 3Sat where the input formula cannot contain a clause with both a negated variable and a non-negated variable. Prove that NoMixedClauses3Sat is NP-hard. *[Hint: Reduce from the standard 3Sat problem.]*

(b) Describe a polynomial-time reduction from NoMixedClauses3Sat to TSA.

7. Prove that the following variants of SAT is NP-hard. *[Hint: Describe reductions from 3SAT.]*

   (a) Given a boolean formula F in conjunctive normal form, where *each variable appears in at most three clauses*, determine whether F has a satisfying assignment. *[Hint: First consider the variant where each variable appears in at most **five** clauses.]*

   (b) Given a boolean formula F in conjunctive normal form *and given one satisfying assignment for F*, determine whether F has at least one other satisfying assignment.

8. Bill Clinton is planning a White House party for his staff. His staff has a hierarchical structure; that is, the supervisor relation forms a directed, rooted, acyclic graph, with Bill at the top, and there is an edge from person i to person j in the graph if and only if i is one of several possible immediate supervisors of person j.

   Vice-president Gore has assigned each White House staff member a "party-hound" rating, which is a nonnegative real number reflecting how likely it is that the person will leave the party wearing a monkey suit and a lampshade. In order to make the party fun for all guests, Bill wants to ensure that if a person i attends, then none of i's immediate supervisors attend also.

   Show that it is NP-hard to determine a guest-list subject to the above constraints, which maximizes the sum of the party-hound ratings of all guests.

   [Hint: This problem can be solved in polynomial time when the input graph is a tree!]

9. Prove that the following problem (which we call Match) is NP-hard. The input is a finite set S of strings, all of the same length n, over the alphabet $\{0, 1, 2\}$. The problem is to determine whether there is a string $w \in \{0, 1\}^n$ such that for every string $s \in S$, the strings s and w have the same symbol in at least one position.

   For example, given the set $S = \{01220, 21110, 21120, 00211, 11101\}$, the correct output is True, because the string $w = 01001$ matches the first three strings of S in the second position, and matches the last two strings of S in the last position. On the other hand, given the set $S = \{00, 11, 01, 10\}$, the correct output is False.

   [Hint: Describe a reduction from SAT (or 3SAT)]

10. To celebrate the end of the semester, Professor Jarling want to treat himself to an ice-cream cone, at the *Polynomial House of Flavors*. For a fixed price, he can build a cone with as many scoops as he'd like. Because he has good balance (and because we want this problem to work out), assume that he can balance any number of scoops on top of the cone without it tipping over. He plans to eat the ice cream one scoop at a time, from top to bottom, and doesn't want more than one scoop of any flavor.

    However, he realizes that eating a scoop of bubblegum ice cream immediately after the scoop of potatoes-and-gravy ice cream would be unpalatable; these two flavors clearly should not be placed next to each other in the stack. He has other similar constraints; certain pairs of flavors cannot be adjacent in the stack.

He'd like to get as much ice cream as he can for the one fee by building the tallest cone possible that meets his flavor-incompatibility constraints. Prove that this problem is NP-hard.

11. At the end of the semester, Professors Erickson and Pitt are forced to solve the following ExamDesign problem. They have a list of problems, and they know for each problem which students will *really enjoy* that problem. They need to choose a subset of problems for the exam such that for each student in the class, the exam includes at least one question that student will really enjoy. On the other hand, the professors do not want to spend the entire summer grading an exam with dozens of questions, so the exam must also contain as few questions as possible. Prove that ExamDesign is NP-hard.

12. Which of the following results would resolve the P vs. NP question? Justify each answer with a short sentence or two.

    (a) The construction of a polynomial time algorithm for some problem in NP.

    (b) A polynomial-time reduction from 3Sat to the language $\{0^n 1^n \mid n \geq 0\}$.

    (c) A polynomial-time reduction from $\{0^n 1^n \mid n \geq 0\}$ to 3Sat.

    (d) A polynomial-time reduction from 3Color to MinVertexCover.

    (e) The construction of a nondeterministic Turing that cannot be simulated by any deterministic Turing without time loss.

Turing Machines and Undecidability

For each of the following languages, either ***sketch*** an algorithm to decide that language or ***prove*** that the language is undecidable, using a diagonal argument, a reduction argument, Rice's theorem, or some combination. Recall that w^R denotes the reversal of string w.

1. $\varnothing$

2. $\{\mathtt{0}^n\mathtt{1}^n\mathtt{2}^n \mid n \ge 0\}$

3. $\{A \in \{\mathtt{0},\mathtt{1}\}^{n\times n} \mid n \ge 0 \text{ and } A \text{ is the adjacency matrix of a dag with } n \text{ vertices}\}$

4. $\{A \in \{\mathtt{0},\mathtt{1}\}^{n\times n} \mid n \ge 0 \text{ and } A \text{ is the adjacency matrix of a 3-colorable graph with } n \text{ vertices}\}$

5. $\{\langle M\rangle \mid M \text{ accepts } \langle M\rangle^R\}$

6. $\{\langle M\rangle \mid M \text{ accepts } \langle M\rangle^R\} \cap \{\langle M\rangle \mid M \text{ rejects } \langle M\rangle^R\}$

7. $\{\langle M\rangle\mathtt{\#}w \mid M \text{ accepts } ww^R\}$

8. $\{\langle M\rangle \mid M \text{ accepts at least one palindrome}\}$

9. $\Sigma^* \setminus \{\langle M\rangle \mid M \text{ accepts at least one palindrome}\}$

10. $\{\langle M\rangle \mid M \text{ rejects at least one palindrome}\}$

11. $\{\langle M\rangle \mid M \text{ accepts exactly one string of length } \ell \text{, for each integer } \ell \ge 0\}$

12. $\{\langle M\rangle \mid \textsc{Accept}(M) \text{ has an infinite fooling set}\}$

13. $\{\langle M\rangle\mathtt{\#}\langle M'\rangle \mid \textsc{Accept}(M) \cap \textsc{Accept}(M') \ne \varnothing\}$

14. $\{\langle M\rangle\mathtt{\#}\langle M'\rangle \mid \textsc{Accept}(M) \oplus \textsc{Reject}(M') \ne \varnothing\}$ — Here $\oplus$ means exclusive-or.

15. $\{\langle M\rangle\mathtt{\#}w \mid M \text{ accepts } w \text{ in at most } 2^{|w|} \text{ steps}\}$

16. $\{\langle M\rangle \mid \text{There is a string } w \text{ that } M \text{ accepts in at most } 2^{|w|} \text{ steps}\}$

*17. $\{\langle M\rangle\mathtt{\#}w \mid M \text{ moves its head to the right on input } w\}$

*18. $\{\langle M\rangle\mathtt{\#}w \mid M \text{ changes a symbol on its tape when given input } w\}$

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?

HamiltonianCycle: Given a graph G (either directed or undirected), is there a cycle in G that visits every vertex exactly once?

HamiltonianPath: Given a graph G (either directed or undirected), is there a path 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?

Partition: Given a set X of positive integers, does X have subsets A and B such that $A \cup B = X$ and $A \cap B = \varnothing$ and $\sum A = \sum B = \frac{1}{2} \sum X$?

SubsetSum: Given a set X of positive integers and a positive integer T, is there a subsets $A \subseteq X$ such that $\sum A = T$?

You may assume the following languages are undecidable:

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