CS/ECE 374 A ✧ Spring 2018

☙ Final Exam Study Questions ❧

This is a "core dump" of potential questions for Midterm 1. This should give you a good idea of the *types* of questions that we will ask on the exam—in particular, there *will* be a series of True/False questions—but the actual exam questions may or may not appear in this handout. This list intentionally includes a few questions that are too long or difficult for exam conditions; these are indicated with a *star.

Don't forget to review the study problems for Midterms 1 and 2; the final exam is cumulative!

☙ How to Use These Problems ❧

Solving every problem in this handout is ***not*** the best way to study for the exam. Memorizing the solutions to every problem in this handout is the ***absolute worst*** way to study for the exam.

What we recommend instead is to work on a *sample* of the problems. Choose one or two problems at random from each section and try to solve them from scratch under exam conditions—by yourself, in a quiet room, with a 30-minute timer, *without* your notes, *without* the internet, and if possible, even without your cheat sheet. If you're comfortable solving a few problems in a particular section, you're probably ready for that type of problem on the exam. Move on to the next section.

Discussing problems with other people (in your study groups, in the review sessions, in office hours, or on Piazza) and/or looking up old solutions can be *extremely* helpful, but ***only after*** you have (1) made a good-faith effort to solve the problem on your own, and (2) you have either a candidate solution or some idea about where you're getting stuck.

If you find yourself getting stuck on a particular type of problem, try to figure out *why* you're stuck. Do you understand the problem statement? Are you stuck on choosing the right high-level approach, are you stuck on the technical details, or are you struggling to express your ideas clearly?

Similarly, if feedback suggests that your solutions to a particular type of problem are incorrect or incomplete, try to figure out what you missed. For induction proofs: Are you sure you have the right induction hypothesis? Are your cases obviously exhaustive? For regular expressions, DFAs, NFAs, and context-free grammars: Is your solution both exclusive and exhaustive? Did you try a few positive examples *and* a few negative examples? For fooling sets: Are you imposing enough structure? Are x and y really *arbitrary* strings from F? For language transformations: Are you transforming in the right direction? Are you using non-determinism correctly? Do you understand the formal notation for DFAs and NFAs?

Remember that your goal is *not* merely to "understand" the solution to any particular problem, but to become more comfortable with solving a certain *type* of problem on your own. **"Understanding" is a trap; aim for mastery.** If you can identify specific steps that you find problematic, read more *about those steps*, focus your practice *on those steps*, and try to find helpful information *about those steps* to write on your cheat sheet. Then work on the next problem!

True or False? (All from previous final exams)

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:

[X] Yes	[ ] No	$2+2=4$
[ ] Yes	[X] No	$x+y=5$
[ ] Yes	[X] No	3SAT can be solved in polynomial time.
[X] Yes	[ ] No	Jeff is not the Queen of England.
[X] Yes	[ ] No	If $P=NP$ then Jeff is the Queen of England.

The actual exam will include forty true/false questions. Each correct choice will be worth ½ point, each incorrect choice will be worth −¼ point, and each **IDK** will be worth +⅛ point.

---

1. Which of the following is a good English specification of a recursive function that can be used to compute the edit distance between two strings $A[1..n]$ and $B[1..n]$?

[ ] Yes	[ ] No	$Edit(i,j)$ is the answer for i and j.
[ ] Yes	[ ] No	$Edit(i,j)$ is the edit distance between $A[i]$ and $B[j]$.
[ ] Yes	[ ] No	$Edit[1..n, 1..n]$ stores the edit distances for all prefixes.
[ ] Yes	[ ] No	$Edit(i,j)$ is the edit distance between $A[i..n]$ and $B[j..n]$.
[ ] Yes	[ ] No	$Edit[i,j]$ is the value stored at row i and column j of the table.

---

2. Which of the following statements are true for ***every*** language $L \subseteq \{0,1\}^*$?

[ ] Yes	[ ] No	L is non-empty.
[ ] Yes	[ ] No	L is infinite.
[ ] Yes	[ ] No	L contains the empty string ε.
[ ] Yes	[ ] No	L^* is infinite.
[ ] Yes	[ ] No	L^* is regular.
[ ] Yes	[ ] No	L is accepted by some DFA if and only if L is accepted by some NFA.

2. (continued) Which of the following are true for ***every*** language $L \subseteq \{0,1\}^*$?

Yes	No	L is described by some regular expression if and only if L is rejected 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 decidable, then L is infinite.
Yes	No	If L is not decidable, then L is infinite.
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	If L has a finite fooling set, then L is decidable.
Yes	No	If L is the union of two regular languages, then its complement $\overline{L}$ is regular.
Yes	No	If L is the union of two regular languages, then its complement $\overline{L}$ is context-free.
Yes	No	If L is the union of two decidable languages, then L is decidable.
Yes	No	If L is the union of two undecidable languages, then L is undecidable.
Yes	No	L is accepted by some NFA with 374 states if and only if L is accepted by some DFA with 374 states.
Yes	No	If $L \notin \mathrm{P}$, then L is not regular.
Yes	No	L is decidable if and only if its complement $\overline{L}$ is undecidable.
Yes	No	Both L and its complement $\overline{L}$ are decidable.

---

3. Which of the following statements are true for ***at least one*** language $L \subseteq \{0,1\}^*$?

Yes	No	L is non-empty.
Yes	No	L is infinite.
Yes	No	L contains the empty string ε.
Yes	No	L^* is finite.

3. (continued) Which of the following are true for ***at least one*** language $L \subseteq \{0,1\}^*$?

Yes	No	L^* is not regular.
Yes	No	L is not regular but L^* is regular.
Yes	No	L is finite and L is undecidable.
Yes	No	L is decidable but L^* is not decidable.
Yes	No	L is not decidable but L^* is decidable.
Yes	No	L is the union of two decidable languages, but L is not decidable.
Yes	No	L is the union of two undecidable languages, but L is decidable.
Yes	No	L is accepted by an NFA with 374 states, but L is not accepted by a DFA with 374 states.
Yes	No	L is accepted by an DFA with 374 states, but L is not accepted by a NFA with 374 states.
Yes	No	L is regular and $L \notin$ P.

---

4. 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 $O(\|w\|^2)$ steps.

---

5. Which of the following languages over the alphabet $\{0, 1\}$ are ***regular***?

Yes	No	$\{0^m 1^n \mid m \geq 0 \text{ and } n \geq 0\}$
Yes	No	All strings with the same number of 0s and 1s
Yes	No	Binary representations of all positive integers divisible by 17
Yes	No	Binary representations of all prime numbers less than 10^{100}
Yes	No	$\{ww \mid w \text{ is a palindrome}\}$
Yes	No	$\{wxw \mid w \text{ is a palindrome and } x \in \{0, 1\}^*\}$
Yes	No	$\{\langle M \rangle \mid M \text{ accepts a finite number of non-palindromes}\}$

---

6. Which of the following languages/decision problems are ***decidable***?

Yes	No	$\emptyset$
Yes	No	$\{0^n 1^{2n} 0^n 1^{2n} \mid n \geq 0\}$
Yes	No	$\{ww \mid w \text{ is a palindrome}\}$
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	$\{\langle M \rangle \# w \mid M \text{ accepts } ww\}$
Yes	No	$\{\langle M \rangle \# w \mid M \text{ accepts } ww \text{ after at most } \lvert w \rvert^2 \text{ transitions}\}$
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'?

---

7. Which of the following languages can be proved undecidable ***using Rice's Theorem***?

Yes	No	$\{0^n 1^{2n} 0^n 1^{2n} \mid n \geq 0\}$
Yes	No	$\{\langle M \rangle \mid M \text{ accepts an infinite number of strings}\}$
Yes	No	$\{\langle M \rangle \mid M \text{ accepts a finite number of strings}\}$
Yes	No	$\{\langle M \rangle \mid M \text{ accepts either } \langle M \rangle \text{ or } \langle M \rangle^R\}$
Yes	No	$\{\langle M \rangle \mid M \text{ accepts both } \langle M \rangle \text{ and } \langle M \rangle^R\}$
Yes	No	$\{\langle M \rangle \mid M \text{ does not accept exactly 374 palindromes}\}$
Yes	No	$\{\langle M \rangle \mid M \text{ accepts some string } w \text{ after at most } \lvert w \rvert^2 \text{ transitions}\}$
Yes	No	$\{\langle M \rangle \# w \mid M \text{ rejects } w \text{ after at most } \lvert w \rvert^2 \text{ transitions}\}$

---

8. Recall the halting language $\text{HALT} = \{\langle M \rangle \# w \mid M \text{ halts on input } w\}$. Which of the following statements about its complement $\overline{\text{HALT}} = \Sigma^* \setminus \text{HALT}$ are true?

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

---

9. 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.
Yes	No	If A has no polynomial-time algorithm then neither does B.

---

10. 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 P = NP.
Yes	No	If B is NP-hard then A is NP-hard.
Yes	No	If B is undecidable then A is undecidable.

---

11. Consider the following pair of languages:

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

    (For concreteness, assume that in both of these languages, graphs are represented by their adjacency matrices.) Which of the following ***must*** be true, assuming P≠NP?

Yes	No	CONNECTED $\in$ NP
Yes	No	HAMPATH $\in$ NP
Yes	No	HAMPATH is decidable.

Yes	No	There is no polynomial-time reduction from HAMPATH to CONNECTED.
Yes	No	There is no polynomial-time reduction from CONNECTED to HAMPATH.

---

12. Consider the following pair of languages:

    - HAMPATH $:= \{G \mid G \text{ is a directed graph with a Hamiltonian path}\}$
    - ACYCLIC $:= \{G \mid G \text{ is a directed acyclic graph}\}$

    (For concreteness, assume that in both of these languages, graphs are represented by their adjacency matrices.) Which of the following ***must*** be true, assuming P≠NP?

Yes	No	ACYCLIC $\in$ NP
Yes	No	ACYCLIC $\cap$ HAMPATH $\in$ P
Yes	No	HAMPATH is decidable.
Yes	No	There is a polynomial-time reduction from HAMPATH to ACYCLIC.
Yes	No	There is a polynomial-time reduction from ACYCLIC to HAMPATH.

---

13. Suppose we want to prove that the following language is undecidable.

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

    Rocket J. Squirrel a reduction from the standard halting language

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

    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 $\langle M\rangle \# w$?

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 $\langle M \rangle \# w$.
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.)

---

14. Suppose we want to prove that the following language is undecidable.

$$\text{MUGGLE} := \left\{ \langle M \rangle \;\middle|\; M \text{ accepts SCIENCE but rejects MAGIC} \right\}$$

Professor Potter, your instructor in Defense Against Models of Computation and Other Dark Arts, suggests a reduction from the standard halting language

$$\text{HALT} := \left\{ \langle M \rangle \# w \;\middle|\; M \text{ halts on inputs } w \right\}.$$

Specifically, suppose there is a Turing machine DETECTOMUGGLETUM that decides MUGGLE. Professor Potter claims that the following algorithm decides HALT.

```
DECIDEHALT(⟨M⟩#w):
  Encode the following Turing machine:
      RUBBERDUCK(x):
        run M on input w
        if x = MAGIC
            return FALSE
        else
            return TRUE
  return DETECTOMUGGLETUM(⟨RUBBERDUCK⟩)
```

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

Yes	No	If M accepts w, then RUBBERDUCK accepts SCIENCE.
Yes	No	If M accepts w, then RUBBERDUCK accepts CHOCOLATE.
Yes	No	If M rejects w, then RUBBERDUCK rejects MAGIC.
Yes	No	If M rejects w, then RUBBERDUCK halts on every input string.
Yes	No	If M diverges on w, then RUBBERDUCK rejects every input string.

Yes	No	If M accepts w, then DETECTOMUGGLETUM accepts ⟨RUBBERDUCK⟩.
Yes	No	If M rejects w, then DECIDEHALT rejects $\langle M \rangle \# w$.
Yes	No	If M diverges on w, then DECIDEHALT rejects $\langle M \rangle \# w$.
Yes	No	DECIDEHALT decides the language HALT. (That is, Professor Potter's reduction is actually correct.)
Yes	No	DECIDEHALT actually runs (or simulates) RUBBERDUCK.
Yes	No	MUGGLE is decidable.

NP-hardness

1. A boolean formula is in *disjunctive normal form* (or *DNF*) if it consists of a *disjunction* (OR) or several *terms*, each of which is the conjunction (AND) of one or more literals. For example, the formula

$$(\overline{x} \wedge y \wedge \overline{z}) \vee (y \wedge z) \vee (x \wedge \overline{y} \wedge \overline{z})$$

is in disjunctive normal form. DNF-SAT asks, given a boolean formula in disjunctive normal form, whether that formula is satisfiable.

   (a) Describe a polynomial-time algorithm to solve DNF-SAT.

   (b) What is the error in the following argument that P=NP?

> *Suppose we are given a boolean formula in conjunctive normal form with at most three literals per clause, and we want to know if it is satisfiable. We can use the distributive law to construct an equivalent formula in disjunctive normal form. For example,*
>
> $$(x \vee y \vee \overline{z}) \wedge (\overline{x} \vee \overline{y}) \iff (x \wedge \overline{y}) \vee (y \wedge \overline{x}) \vee (\overline{z} \wedge \overline{x}) \vee (\overline{z} \wedge \overline{y})$$
>
> *Now we can use the algorithm from part (a) to determine, in polynomial time, whether the resulting DNF formula is satisfiable. We have just solved* 3Sat *in polynomial time. Since* 3Sat *is NP-hard, we must conclude that P=NP!*

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.

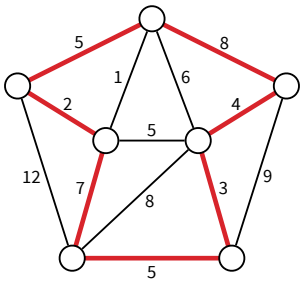
   (a) Give an example of a graph that contains a infra-Hamiltonian cycle, but does not contain a Hamiltonian cycle (which visits every vertex exactly once).

   (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. A subset S of vertices in an undirected graph G is ***almost independent*** if at most 374 edges in G have both endpoints in S. Prove that finding the size of the largest almost-independent set of vertices in a given undirected graph is NP-hard.

7. Let G be an undirected graph with weighted edges. A *heavy Hamiltonian cycle* is a cycle C that passes through each vertex of G exactly once, such that the total weight of the edges in C is more than half of the total weight of all edges in G. Prove that deciding whether a graph has a heavy Hamiltonian cycle is NP-hard.



A heavy Hamiltonian cycle. The cycle has total weight 34; the graph has total weight 67.

8. (a) A *tonian path* in a graph G is a path that goes through at least half of the vertices of G. Show that determining whether a graph has a tonian path is NP-hard.

   (b) A *tonian cycle* in a graph G is a cycle that goes through at least half of the vertices of G. Show that determining whether a graph has a tonian cycle is NP-hard. *[Hint: Use part (a). Or not.]*

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

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

10. 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.

11. The president of Sham-Poobanana University is planning a Mardi Gras party for the university staff. His staff has a hierarchical structure; that is, the supervisor relation forms a directed, acyclic graph, with the president as the only source, and there is an edge from person i to person j in the graph if and only if person i is an immediate supervisor of person j. (Many people on the staff have multiple positions, and thus have several immediate supervisors.) In order to make the party fun for all guests, the president wants to ensure that if a person i attends, then none of i's immediate supervisors can attend.

By mining each staff member's email and social media accounts, Sham-Poobanana University Human Resources has determined a "party-hound" rating for each staff member, which is a non-negative real number reflecting how likely it is that the person will leave the party wearing a monkey suit and a lampshade.

Show that it is NP-hard to determine a guest-list that *maximizes* the sum of the party-hound ratings of all invited guests, subject to the supervisor constraint.

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

12. 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)]

13. 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.

14. At the end of every semester, Jeff needs to solve the following ExamDesign problem. He has a list of problems, and he knows for each problem which students will *really enjoy* that problem. He needs 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, he does 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 the ExamDesign problem is NP-hard.

15. 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 machine that cannot be simulated by any deterministic Turing machine with the same running time.

Some useful NP-hard problems. You are welcome to use any of these in your own NP-hardness proofs, except of course for the specific problem you are trying to prove 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 distinct 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?

MinSetCover: Given a collection of subsets $S_1, S_2, \ldots, S_m$ of a set S, what is the size of the smallest subcollection whose union is S?

MinHittingSet: Given a collection of subsets $S_1, S_2, \ldots, S_m$ of a set S, what is the size of the smallest subset of S that intersects every subset S_i?

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

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

LongestPath: Given a graph G (either directed or undirected, possibly with weighted edges), what is the length of the longest simple path in G?

SteinerTree: Given an undirected graph G with some of the vertices marked, what is the minimum number of edges in a subtree of G that contains every marked vertex?

SubsetSum: Given a set X of positive integers and an integer k, does X have a subset whose elements sum to k?

Partition: Given a set X of positive integers, can X be partitioned into two subsets with the same sum?

3Partition: Given a set X of $3n$ positive integers, can X be partitioned into n three-element subsets, all with the same sum?

IntegerLinearProgramming: Given a matrix $A \in \mathbb{Z}^{n \times d}$ and two vectors $b \in \mathbb{Z}^n$ and $c \in Z^d$, compute $\max\{c \cdot x \mid Ax \leq b, x \geq 0, x \in \mathbb{Z}^d\}$.

FeasibleILP: Given a matrix $A \in \mathbb{Z}^{n \times d}$ and a vector $b \in \mathbb{Z}^n$, determine whether the set of feasible integer points $\max\{x \in \mathbb{Z}^d \mid Ax \leq b, x \geq 0\}$ is empty.

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?

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

SteamedHams: Aurora borealis? At this time of year, at this time of day, in this part of the country, localized entirely within your kitchen? May I see it?

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 diagonalization argument, a reduction argument, Rice's theorem, closure properties, or some combination of the above. Recall that w^R denotes the reversal of string w.

1. $\emptyset$
2. $\{\mathtt{0}^n\mathtt{1}^n\mathtt{2}^n \mid n \geq 0\}$
3. $\{A \in \{\mathtt{0},\mathtt{1}\}^{n\times n} \mid n \geq 0 \text{ and } A \text{ is the adjacency matrix of a dag with } n \text{ vertices}\}$
4. $\{A \in \{\mathtt{0},\mathtt{1}\}^{n\times n} \mid n \geq 0 \text{ and } A \text{ is the adjacency matrix of a 3-colorable graph with } n \text{ vertices}\}$
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. $\{\langle M\rangle \# w \mid M \text{ accepts } ww^R\}$
8. $\{\langle M\rangle \mid M \text{ accepts } \texttt{RICESTHEOREM}\}$
9. $\{\langle M\rangle \mid M \text{ rejects } \texttt{RICESTHEOREM}\}$
10. $\{\langle M\rangle \mid M \text{ accepts at least one palindrome}\}$
11. $\Sigma^* \setminus \{\langle M\rangle \mid M \text{ accepts at least one palindrome}\}$
12. $\{\langle M\rangle \mid M \text{ rejects at least one palindrome}\}$
13. $\{\langle M\rangle \mid M \text{ accepts exactly one string of length } \ell, \text{ for each integer } \ell \geq 0\}$
14. $\{\langle M\rangle \mid \textsc{Accept}(M) \text{ has an infinite fooling set}\}$
15. $\{\langle M\rangle \# \langle M'\rangle \mid \textsc{Accept}(M) \cap \textsc{Accept}(M') \neq \emptyset\}$
16. $\{\langle M\rangle \# \langle M'\rangle \mid \textsc{Accept}(M) \oplus \textsc{Reject}(M') \neq \emptyset\}$ — Here $\oplus$ means exclusive-or.
17. $\{\langle M\rangle \# w \mid M \text{ accepts } w \text{ in at most } 2^{|w|} \text{ steps}\}$
18. $\{\langle M\rangle \mid \text{There is a string } w \text{ that } M \text{ accepts in at most } 2^{|w|} \text{ steps}\}$

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

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

Some useful undecidable problems. You are welcome to use any of these in your own undecidability proofs, except of course for the specific problem you are trying to prove undecidable.

$$\textsc{SelfReject} := \{\langle M\rangle \mid M \text{ rejects } \langle M\rangle\}$$
$$\textsc{SelfAccept} := \{\langle M\rangle \mid M \text{ accepts } \langle M\rangle\}$$
$$\textsc{SelfHalt} := \{\langle M\rangle \mid M \text{ halts on } \langle M\rangle\}$$
$$\textsc{SelfDiverge} := \{\langle M\rangle \mid M \text{ does not halt on } \langle M\rangle\}$$
$$\textsc{Reject} := \{\langle M\rangle \# w \mid M \text{ rejects } w\}$$
$$\textsc{Accept} := \{\langle M\rangle \# w \mid M \text{ accepts } w\}$$
$$\textsc{Halt} := \{\langle M\rangle \# w \mid M \text{ halts on } w\}$$
$$\textsc{Diverge} := \{\langle M\rangle \# w \mid M \text{ does not halt on } w\}$$
$$\textsc{NeverReject} := \{\langle M\rangle \mid \textsc{Reject}(M) = \emptyset\}$$
$$\textsc{NeverAccept} := \{\langle M\rangle \mid \textsc{Accept}(M) = \emptyset\}$$
$$\textsc{NeverHalt} := \{\langle M\rangle \mid \textsc{Halt}(M) = \emptyset\}$$
$$\textsc{NeverDiverge} := \{\langle M\rangle \mid \textsc{Diverge}(M) = \emptyset\}$$