

CS/ECE 374 A ✧ Fall 2019

Final Exam

December 13, 2019

Real name:	
NetID:	

Gradescope name:	
Gradescope email:	

-
- **Don't panic!**
 - If you brought anything except your writing implements and your two double-sided $8\frac{1}{2} \times 11$ " cheat sheets, please put it away for the duration of the exam. In particular, please turn off and put away *all* medically unnecessary electronic devices.
 - Please clearly print your real name, your university NetID, your Gradescope name, and your Gradescope email address in the boxes above. **We will not scan this page into Gradescope.**
 - Please also print **only the name you are using on Gradescope** at the top of every page of the answer booklet, except this cover page. These are the pages we will scan into Gradescope.
 - Please do not write outside the black boxes on each page; these indicate the area of the page that the scanner can actually see.
 - **Please read the entire exam before writing anything.** Please ask for clarification if any question is unclear.
 - **The exam lasts 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.
 - 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".
 - **Please return your cheat sheets and all scratch paper with your answer booklet.**
 - May the Sith be with you.
-

Beware of the man who works hard to learn something,
learns it, and finds himself no wiser than before.

He is full of murderous resentment of people who are ignorant
without having come by their ignorance the hard way.

— Bokonon

Final Exam Problem 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, mark the “No” box. For example:

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

There are 40 yes/no choices altogether. Each correct choice is worth $+1/2$ point; each incorrect choice is worth $-1/4$ point; each checked “IDK” is worth $+1/8$ point.

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

Yes	No	IDK	L^* is non-empty.
Yes	No	IDK	L^* is regular.
Yes	No	IDK	L^* is decidable.
Yes	No	IDK	If L is NP-hard, then L is not regular.
Yes	No	IDK	If L is not regular, then L is undecidable.
Yes	No	IDK	If L is context-free, then L is infinite.
Yes	No	IDK	L is the intersection of two regular languages if and only if L is regular.
Yes	No	IDK	L is decidable if and only if L^* is decidable.
Yes	No	IDK	L is decidable if and only if its reversal $L^R = \{w^R \mid w \in L\}$ is decidable. (Recall that w^R denotes the reversal of the string w .)
Yes	No	IDK	L is decidable if and only if its complement \bar{L} is undecidable.

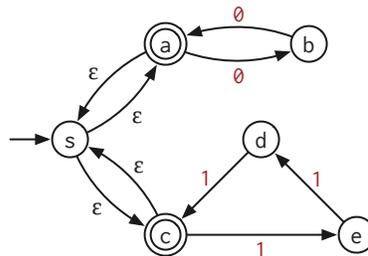
(b) Consider the following sets of undirected graphs:

- TREES is the set of all connected undirected graphs with no cycles.
- 3COLOR is the set of all undirected graphs that can be properly colored using at most 3 colors.

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

Yes	No	IDK	$TREES \in NP$
Yes	No	IDK	$TREES \subseteq 3COLOR$
Yes	No	IDK	There is a polynomial-time reduction from TREES to 3COLOR
Yes	No	IDK	There is a polynomial-time reduction from 3COLOR to TREES
Yes	No	IDK	TREES is NP-hard.

(c) Let M be the following NFA:



Which of the following statements about M are true?

Yes	No	IDK	M accepts the empty string ϵ
Yes	No	IDK	$\delta^*(s, 010) = \{s, a, c\}$
Yes	No	IDK	$\epsilon\text{-reach}(a) = \{s, a, c\}$
Yes	No	IDK	M rejects the string 11100111000
Yes	No	IDK	$L(M) = (00)^* + (111)^*$

(d) Which of the following languages over the alphabet $\Sigma = \{0, 1\}$ are **regular**? Recall that $\#(a, w)$ denotes the number of times symbol a appears in string w .

- | | | | |
|-----|----|-----|---|
| Yes | No | IDK | The intersection of two regular languages |
| Yes | No | IDK | $\{w \in \Sigma^* \mid w \text{ is prime}\}$ |
| Yes | No | IDK | $\{w \in \Sigma^* \mid \#(0, w) + \#(1, w) > 374\}$ |
| Yes | No | IDK | $\{w \in \Sigma^* \mid \#(0, w) - \#(1, w) > 374\}$ |
| Yes | No | IDK | The language generated by the context-free grammar $S \rightarrow 0S \mid 10S \mid \varepsilon$ |

(e) Which of the following languages or problems are **decidable**?

- | | | | |
|-----|----|-----|--|
| Yes | No | IDK | Σ^* |
| Yes | No | IDK | 3SAT |
| Yes | No | IDK | $\{\langle M \rangle \mid M \text{ accepts every string whose length is prime}\}$ |
| Yes | No | IDK | $\{\langle M \rangle \mid M \text{ accepts all strings in } 0^* \text{ and rejects all strings in } 1^*\}$ |
| Yes | No | IDK | $\{\langle M \rangle \mid M \text{ is a Turing machine with at least two states}\}$ |

(f) Which of the following languages or problems can be proved undecidable **using Rice's Theorem**?

- | | | | |
|-----|----|-----|--|
| Yes | No | IDK | Σ^* |
| Yes | No | IDK | 3SAT |
| Yes | No | IDK | $\{\langle M \rangle \mid M \text{ accepts every string whose length is prime}\}$ |
| Yes | No | IDK | $\{\langle M \rangle \mid M \text{ accepts all strings in } 0^* \text{ and rejects all strings in } 1^*\}$ |
| Yes | No | IDK | $\{\langle M \rangle \mid M \text{ is a Turing machine with at least two states}\}$ |

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

$$\text{MARVIN} := \{ \langle M \rangle \mid M \text{ rejects an infinite number of strings} \}$$

Professor Prefect, your instructor in Vogon Poetry and Knowing Where Your Towel Is, suggests a reduction from the standard halting language

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

Specifically, suppose there is a program PARANOIDANDROID that decides MARVIN. Professor Prefect claims that the following algorithm decides HALT.

```
DECIDEHALT( $\langle M \rangle, w$ ):  
  Write code for the following algorithm:  
    HOOPYFROOD( $x$ ):  
      run  $M$  on input  $w$   
      if  $x = \text{DONT PANIC}$   
        return TRUE  
      else  
        return FALSE  
  return PARANOIDANDROID( $\langle \text{HOOPYFROOD} \rangle$ )
```

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

- | | | | |
|--------------------------|--------------------------|--------------------------|---|
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | If M accepts w , then HOOPYFROOD accepts BEEBLEBROX. |
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | If M rejects w , then HOOPYFROOD rejects BEEBLEBROX. |
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | If M hangs on w , then HOOPYFROOD rejects DONT PANIC. |
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | PARANOIDANDROID accepts $\langle \text{HOOPYFROOD} \rangle$. |
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | DECIDEHALT decides HALT; that is, Professor Prefect's proof is correct. |

Final Exam Problem 2

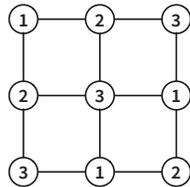
You are planning a hiking trip in Jellystone National Park over winter break. You have a complete map of the park's trails; the map indicates that some trail segments have a high risk of bear encounters. All visitors to the park are required to purchase a canister of bear repellent. You can safely traverse a high-bear-risk trail segment only by *completely* using up a *full* canister of bear repellent. The park rangers have installed refilling stations at several locations around the park, where you can refill empty canisters at no cost. The canisters themselves are expensive and heavy, so you cannot carry more than one. Because the trails are narrow, each trail segment allows traffic in only one direction.

You have converted the trail map into a directed graph $G = (V, E)$, whose vertices represent trail intersections, and whose edges represent trail segments. A subset $R \subseteq V$ of the vertices indicate the locations of the *Repellent Refilling stations*, and a subset $B \subseteq E$ of the edges are marked as having a high risk of Bears. Your campsite appears on the map as a particular vertex $s \in V$, and the visitor center is another vertex $t \in V$.

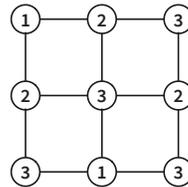
- (a) Describe and analyze an algorithm to decide if you can safely walk from your campsite s to the visitor center t . Assume there is a refill station at your camp site, and another refill station at the visitor center.
 - (b) Describe and analyze an algorithm to decide if you can walk safely from any refill station any other refill station. In other words, for *every* pair of vertices u and v in R , is there a safe path from u to v ?
-

Final Exam Problem 3

Recall that a *proper 3-coloring* of a graph G assigns each vertex of G one of three colors, so that every edge of G has endpoints with different colors. A proper 3-coloring is *balanced* if each color is assigned to *exactly* the same number of vertices.



A balanced proper 3-coloring.



A proper 3-coloring that is not balanced.

The `BALANCED3COLOR` problem asks, given an undirected graph G , whether G has a balanced proper 3-coloring. **Prove** that `BALANCED3COLOR` is NP-hard.

Final Exam Problem 4

For each of the following languages, state whether the language is regular or not, and then justify your answer as follows:

- If the language is regular, *either* give an regular expression that describes the language, *or* draw/describe a DFA or NFA that accepts the language. You do not need to prove that your automaton or regular expression is correct.
- If the language is not regular, *prove* that the language is not regular.

[Hint: Exactly one of these languages is regular.]

- (a) $\{xy \mid x, y \in \Sigma^+ \text{ and } x \text{ and } y \text{ are both palindromes}\}$
- (b) $\{xy \mid x, y \in \Sigma^+ \text{ and } x \text{ is not a palindrome}\}$
-

Final Exam Problem 5

- (a) Recall that a *palindrome* is any string that is equal to its reversal, like REDIVIDER or POOP. Describe an algorithm to find the length of the longest subsequence of a given string that is a palindrome.
- (b) A *double palindrome* is the concatenation of two *non-empty* palindromes, like POOPREDIVIDER or POOPPOOP. Describe an algorithm to find the length of the longest subsequence of a given string that is a *double* palindrome. [Hint: Use your algorithm from part (a).]

For both algorithms, the input is an array $A[1..n]$, and the output is an integer. For example, given the string MAYBEDYNAMICPROGRAMMING as input, your algorithm for part (a) should return 7 (for the palindrome subsequence NMRORMN), and your algorithm for part (b) should return 12 (for the double palindrome subsequence MAYBYAMIRORI).

CS/ECE 374 A ✧ Fall 2019

Gradescope name:

Final Exam Problem 6

Let M be an arbitrary DFA. Describe and analyze an efficient algorithm to decide whether M rejects an infinite number of strings.

(scratch paper)

(scratch paper)

(scratch paper)

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, \dots, 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, \dots, 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 $c \in \mathbb{Z}^d$, 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?