CS/ECE 374 A ✧ Fall 2019

❦ Final Exam ❦

December 13, 2019

Real name:	Dakshita
NetID:	

Gradescope name:	
Gradescope email:	

- ***Don't panic!***
- 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, 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

CS/ECE 374 A ✧ Fall 2019 **Final Exam Problem 1**	Gradescope name:

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

There are **40** yes/no choices altogether. Each correct choice is worth +½ point; each incorrect choice is worth −¼ point; each checked "IDK" is worth +1/8 point.

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

check this what is L^* for $L=\phi$

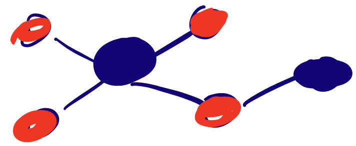
Yes	~~No~~ X	IDK	L^* is non-empty.	$L=\phi$, $L^*=\phi$
Yes	~~No~~ X	IDK	L^* is regular.	$L=\{0^{2^i}1 \mid i>0\}$ $L^*=0^{2^{i_1}}10^{2^{i_2}}1\ldots$
Yes	~~No~~ X	IDK	L^* is decidable.	$L=\{\langle M\rangle, x \mid M \text{ halts on } x\}$
~~Yes~~ X	No	IDK	If L is NP-hard, then L is not regular.	L is regular $\Rightarrow$ L in not NP hard. L is in P.
Yes	~~No~~ X	IDK	If L is not regular, then L is undecidable.	$\{0^n1^n \mid n>0\}$
Yes	~~No~~ X	IDK	If L is context-free, then L is infinite.	$\{\varepsilon\}$
~~Yes~~ X	No	IDK	L is the intersection of two regular languages if and only if L is regular.	by definition $\Rightarrow$; $L \cap L \Leftarrow L$
Yes	~~No~~ X	IDK	L is decidable if ~~and~~ $\not\Leftarrow$ only if L^* is decidable.	L^* is Σ^* which is decidable; $L=\{\langle M\rangle, x \mid M \text{ halts on } x\}\cup\{0,1\}$ is undecidable
~~Yes~~ X	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~~ X	IDK	L is decidable if and only if its complement $\overline{L}$ is undecidable.	$L=\phi$, $\overline{L}=\Sigma^*$

(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≠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.



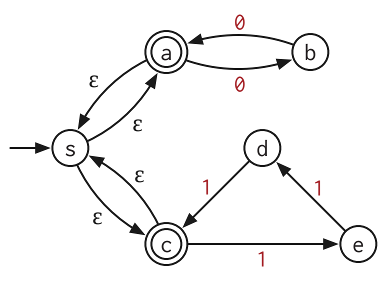
TREES is poly-time solvable.

$\in$ P

NP-hard

no, TREES $\in$ P

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

$\delta^*(s, 010) = \emptyset$

$00111 \notin (00)^* + (111)^*$

$(00+111)^*$

(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	$\{w \in \Sigma^* \mid \#(0, w) - \#(1, w) > \cancel{374}\}$ n^2 $0^n 1^{n-3}$ $0^n 1^n$
~~Yes~~	No	IDK	The language generated by the context-free grammar $S \to 0S \mid 10S \mid \varepsilon$ $(0+10)^*$

(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^* decidable
Yes	~~No~~	IDK	3SAT decidable
~~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	$\{\langle M \rangle \mid M \text{ is a Turing machine with at least two states}\}$

Y = accept everything
N = reject everything
this is the empty language $L = \emptyset$.

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

$$\textsc{Marvin} := \big\{ \langle M \rangle \;\big|\; M \text{ rejects an infinite number of strings} \big\}$$

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

$$\textsc{Halt} := \big\{ (\langle M \rangle, w) \;\big|\; M \text{ halts on inputs } w \big\}.$$

Specifically, suppose there is a program ParanoidAndroid that decides Marvin. Professor Prefect claims that the following algorithm decides Halt.

```
DecideHalt(⟨M⟩, w):
    Write code for the following algorithm:
        HoopyFrood(x):
            run M on input w
            if x = DONTPANIC
                return True
            else
                return False
    return ParanoidAndroid(⟨HoopyFrood⟩)
```

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

Yes	No	IDK	If M accepts w, then HoopyFrood accepts BEEBLEBROX.
Yes	No	IDK	If M rejects w, then HoopyFrood rejects BEEBLEBROX.
Yes	No	IDK	If M hangs on w, then HoopyFrood rejects DONTPANIC.
Yes	No	IDK	ParanoidAndroid accepts $\langle$HoopyFrood$\rangle$.
Yes	No	IDK	DecideHalt decides Halt; that is, Professor Prefect's proof is correct.

CS/ECE 374 A ✧ Fall 2019 Final Exam Problem 2	Gradescope name:

Graph

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 *R*epellent *R*efilling stations, and a subset $B \subseteq E$ of the edges are marked as having a high risk of *B*ears. 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.

any refill station   any other station

(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?

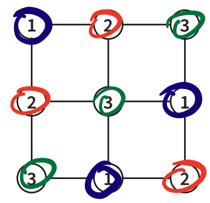
(See problem 2 on HW9)

2 (continued)

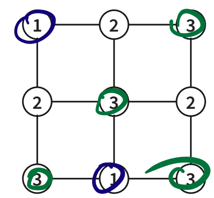
2 (continued)

CS/ECE 374 A ✧ Fall 2019 Final Exam Problem 3	Gradescope name:

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.

Known NP-hard problem: 3COLOR

We will provide a polynomial-time reduction.

from 3COLOR $\leq_P$ BALANCED3COLOR, as follows.
to

On input graph $G=(V,E)$, the reduction makes 3 copies of G, call them $G_1=(V_1,E_1)$, $G_2=(V_2,E_2)$, $G_3=(V_3,E_3)$.

Then Graph $G'=(V',E')$ where

$V' = V_1 \cup V_2 \cup V_3$, [The reduction clearly runs in polynomial (actually $O(|V|+|E|)$ time]

$E' = E_1 \cup E_2 \cup E_3$.

Claim. G is 3-colorable $\Longleftrightarrow$ G' has a balanced 3-coloring.

Proof of Claim: 1 G is 3-colorable

$\Rightarrow G_1$ is 3-colorable.

Let the 3 colors be R, B, Y without loss of generality.

For every vertex colored R in G_1, assign its clone in G_2 the color B, and its clone in G_3 the color Y.

For every vertex colored B in G_1, assign its clone in G_2 the color Y, and its clone in G_3 the color R.

For every vertex colored Y in G_1, assign its clone in G_2 the color R, and its clone in G_3 the color B.

Exactly a third of the vertices in G have color R/Y/B.

$\Rightarrow G'$ is balanced 3-colorable.

2 G' is (balanced) 3-colorable $\Rightarrow G_1$ is 3-colorable
$\Rightarrow G$ is 3-colorable.

CS/ECE 374 A ✧ Fall 2019 Final Exam Problem 4	Gradescope name:

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

Regular

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

(a) $\{xy \mid x$ and y are both palindromes$\}$ Not regular

$F = \{0^*1\}$

$x = 0^i1$ $\quad y = 0^j1$ $\quad z = 0^i$

$xz = \underline{0^i10^i}$ $\quad yz = \underline{0^j10^i}$ $\quad j > i$

$\notin L$ $\quad \underline{0^{j-i}}\,\underline{0^i10^i} \in L$.

(b) Regular $\quad \underbrace{1}\underbrace{1} \notin L$, 1 is a palindrome $\boxed{101}$ → not a palindrome $\in L$

$L = \{xy \mid x, y \in \Sigma^+$ and x is not a palindrome$\}$.

$\underline{11111\ldots}$ ( ) $\quad \underbrace{11}\underbrace{1011}\ldots$

$00\ldots$ $\quad \underbrace{001} \quad \underbrace{01}$

$L = \{z \mid$ there is a 01 or 10 in the first n-1 bits of $z\}$ where $|z| = n$.

$= (0+1)^*(01+10)(0+1)^*(0^4+1)$.

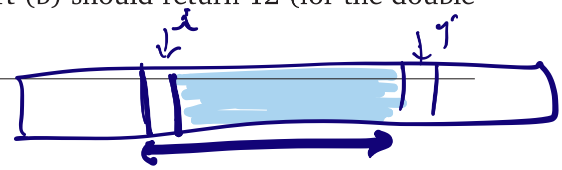
CS/ECE 374 A ✧ Fall 2019 Final Exam Problem 5	Gradescope name:

D.P.

(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 (length). 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).



$A[1 \ldots n]$

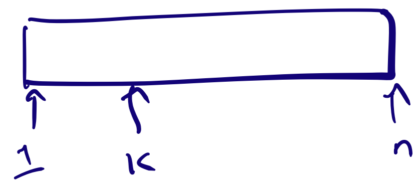
Longps$[i,j]$ denotes the longest palindrome subsequence in the sub-array $A[i \ldots j]$

$$\text{Longps}[i,j] = \begin{cases} 0 & \text{if } i > j \\ 1 & \text{if } i = j \\ \max\begin{pmatrix} \text{Longps}[i+1,j-1]+2 \\ \text{Longps}[i,j-1] \\ \text{Longps}[i+1,j] \end{pmatrix} & \text{if } A[i] = A[j] \\ \max\begin{pmatrix} \text{Longps}[i,j-1] \\ \text{Longps}[i+1,j] \end{pmatrix} & \text{if } A[i] \neq A[j] \end{cases}$$

2-D array LPS. Fill this in decreasing order of i and increasing order of j.

Time $O(n^2)$

(b) $\max_{k \in \{1 \ldots n\}} LPS[1,k] + LPS[k,n]$



This can be computed in $O(n)$ time given LPS.

Total time $= O(n^2) + O(n) = O(n^2)$.

CS/ECE 374 A ✧ Fall 2019	Gradescope name:
Final Exam Problem 6	

Graphs

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

M rejects an infinite number of strings iff there is a cycle in M that reaches a non-accepting state, and is reachable from the start state.

$M = (Q, s, A, \delta) \rightarrow$ directed graph. $V = Q$
E = state transitions
$|E| = 2|Q|$

* Remove states unreachable from the start state in this graph. Using whatever-f-s in $O(|V|+|E|)$ time. Say FALSE if only accepting states remain.

* Add a new vertex t, add outgoing edges from all rejecting states to t.

* Remove vertices not reachable from t, Using whatever-f-s (on reverse graph) in $O(V+E)$ time.

* True if the resulting graph has a cycle, FALSE otherwise.

(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 $S_1, S_2, \ldots, S_m$ 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?