CS/ECE 374 A ✧ Fall 2019

Fake Midterm 1

September 30, 2019

Real name:	
NetID:	

Gradescope name:	
Gradescope email:	

- ***Don't panic!***
- If you brought anything except your writing implements, your **hand-written** double-sided 8½" × 11" cheat sheet, and your university ID, 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. However, if you are using your real name and your university email address on Gradescope, you do ***not*** need to write everything twice. **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.
- If you run out of space for an answer, feel free to use the scratch pages at the back of the answer booklet, but **please clearly indicate where we should look**.
- Proofs are required for full credit if and only if we explicitly ask for them, using the word ***prove*** in bold italics.
- Please return ***all*** paper with your answer booklet: your question sheet, your cheat sheet, and all scratch paper.

CS/ECE 374 A ✧ Fall 2019 Fake Midterm 1 Problem 1	Gradescope name:

For each statement below, check "True" if the statement is ***always*** true and "False" otherwise. Each correct answer is worth +1 point; each incorrect answer is worth −½ point; checking "I don't know" is worth +¼ point; and flipping a coin is (on average) worth +¼ point.

Yes No IDK — Every integer in the empty set is prime.

Yes No IDK — The language $\{0^m 1^n \mid m + n \leq 374\}$ is regular.

Yes No IDK — The language $\{0^m 1^n \mid m - n \leq 374\}$ is regular.

Yes No IDK — For all languages L, the language L^* is regular.

Yes No IDK — For all languages L, the language L^* is infinite.

Yes No IDK — For all languages $L \subset \Sigma^*$, if L can be represented by a regular expression, then $\Sigma^* \setminus L$ is recognized by a DFA.

Yes No IDK — For all languages L and L', if $L \cap L' = \varnothing$ and L' is not regular, then L is regular.

Yes No IDK — Let $M = (\Sigma, Q, s, A, \delta)$ and $M' = (\Sigma, Q, s, Q \setminus A, \delta)$ be arbitrary **DFA**s with identical alphabets, states, starting states, and transition functions, but with complementary accepting states. Then $L(M) \cap L(M') = \varnothing$.

Yes No IDK — Let $M = (\Sigma, Q, s, A, \delta)$ and $M' = (\Sigma, Q, s, Q \setminus A, \delta)$ be arbitrary **NFA**s with identical alphabets, states, starting states, and transition functions, but with complementary accepting states. Then $L(M) \cap L(M') = \varnothing$.

Yes No IDK — For all context-free language L, the language L^* is also context-free.

CS/ECE 374 A ✧ Fall 2019 Fake Midterm 1 Problem 2	Gradescope name:

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. ***Exactly one of these two languages is regular.*** Both of these languages contain the string 00110100000110100.

1. $\left\{0^n w 0^n \mid w \in \Sigma^+ \text{ and } n > 0\right\}$

2. $\left\{w 0^n w \mid w \in \Sigma^+ \text{ and } n > 0\right\}$

CS/ECE 374 A ✧ Fall 2019 Fake Midterm 1 Problem 3	Gradescope name:

The *parity* of a bit-string w is `0` if w has an even number of `1`s, and `1` if w has an odd number of `1`s. For example:

$$parity(\varepsilon) = \mathtt{0} \qquad parity(\mathtt{0010100}) = \mathtt{0} \qquad parity(\mathtt{00101110100}) = \mathtt{1}$$

(a) Give a *self-contained*, formal, recursive definition of the *parity* function. (In particular, do ***not*** refer to # or other functions defined in class.)

(b) Let L be an arbitrary regular language. Prove that the language $OddParity(L) := \{w \in L \mid parity(w) = \mathtt{1}\}$ is also regular.

(c) Let L be an arbitrary regular language. Prove that the language $AddParity(L) := \{parity(w) \cdot w \mid w \in L\}$ is also regular.

[Hint: Yes, you have enough room.]

CS/ECE 374 A ✧ Fall 2019 Fake Midterm 1 Problem 4	Gradescope name:

For each of the following languages L, give a regular expression that represents L ***and*** describe a DFA that recognizes L. You do ***not*** need to prove that your answers are correct.

(a) All strings in $(0+1)^*$ that do not contain the substring 0110.

(b) All strings in 0^*10^* whose length is a multiple of 3.

CS/ECE 374 A ✧ Fall 2019 Fake Midterm 1 Problem 5	Gradescope name:

Consider the language L of all strings in $\{0, 1\}^*$ in which the number of 0s is even, the number of 1s is divisible by 3, and the total number of symbols is divisible by 5. For example, the strings 01011 and 0000000000 are in L, but the strings 01001 and 10101010 are not.

Formally describe a DFA $M = (Q, s, A, \delta)$ over the alphabet $\Sigma = \{0, 1\}$ that recognizes L. ***Do not attempt to draw the DFA. Do not use the phrase "product construction".*** Instead, formally and explicitly specify each of the the components Q, s, A, and δ.

Write your answers in the separate answer booklet.
Please return this question sheet and your cheat sheet with your answers.

1. For each statement below, check "Yes" if the statement is ***always*** true and "No" otherwise. Each correct answer is worth +1 point; each incorrect answer is worth −½ point; checking "I don't know" is worth +¼ point; and flipping a coin is (on average) worth +¼ point. You do ***not*** need to prove your answer is correct.

   Read each statement *very* carefully. Some of these are deliberately subtle.

   (a) If $2+2=5$, then zero is odd.

   (b) Language L is regular if and only if there is a DFA that accepts every string in L.

   (c) Two languages L and L' are regular if and only if $L \cup L'$ is regular.

   (d) For every language L, if L^* is empty, then L is empty.

   (e) Every regular language is recognized by a DFA with exactly one accepting state.

   (f) If L has a fooling set of size 374, then L is regular.

   (g) The language $\left\{0^{374n} \mid n \geq 374\right\}$ is regular.

   (h) The language $\left\{0^{37n}1^{4n} \mid n \geq 374\right\}$ is regular.

   (i) The language $\left\{0^{3n}1^{74n} \mid n \leq 374\right\}$ is regular.

   (j) Every language is either regular or context-free.

2. For any string $w \in \{0,1\}^*$, let $slash(w)$ be the string in $\{0,1,/\}^*$ obtained from w by inserting a new symbol / between any two consecutive appearances of the same symbol. For example:

$$slash(\varepsilon) = \varepsilon$$
$$slash(10101) = 10101$$
$$slash(001010111) = 0/010101/1/1$$

   For any language $L \subseteq \{0,1\}^*$, let $slash(L) = \{slash(w) \mid w \in L\}$.

   (a) Draw or describe a DFA that accepts the language $slash(\{0,1\}^*)$.

   (b) Give a regular expression for the language $slash(\{0,1\}^*)$.

   (c) ***Prove*** that for any regular language L, the language $slash(L)$ is also regular.

   (You do not need to justify your answers to parts (a) and (b).)

3. Let L be the language $\{0^a 1^b 0^c \mid 2a = b + c\}$.

   (a) ***Prove*** that L is not a regular language.

   (b) Describe a context-free grammar for L. (You do not need to justify your answer.)

4. For each of the following languages L, give a regular expression that represents L ***and*** draw or describe a DFA that recognizes L. You do not need to justify your answers.

   (a) All strings in $\{0, 1\}^*$ that do not contain either 100 or 011 as a substring

   (b) All strings in $\{0, 1, 2\}^*$ that do not contain either 01 or 12 or 20 as a substring

5. For any string $w \in \{0, 1\}^*$, let *stupefy*(w) denote the string obtained from w by deleting the first 1 (if any) and replacing each remaining 1 with a 0. For example:

$$\begin{aligned}
stupefy(\varepsilon) &= \varepsilon \\
stupefy(000) &= 000 \\
stupefy(00100) &= 0000 \\
stupefy(111111) &= 00000 \\
stupefy(0100001101) &= 000000000
\end{aligned}$$

   Let L be an arbitrary regular language.

   (a) ***Prove*** that the language $\{stupefy(w) \mid w \in L\}$ is regular.

   (b) ***Prove*** that the language $\{w \in \{0, 1\}^* \mid stupefy(w) \in L\}$ is regular.

Write your answers in the separate answer booklet.
Please return this question sheet and your cheat sheet with your answers.

1. For each statement below, check "Yes" if the statement is ***always*** true and "No" otherwise. Each correct answer is worth +1 point; each incorrect answer is worth −½ point; checking "I don't know" is worth +¼ point; and flipping a coin is (on average) worth +¼ point. You do ***not*** need to prove your answer is correct.

   Read each statement *very* carefully. Some of these are deliberately subtle.

   (a) If zero is odd, then $2 + 2 = 5$.

   (b) For every language L, and for every string $w \in L$, there is a DFA that accepts w.

   (c) Two languages L and L' are regular if and only if $L \cap L'$ is regular.

   (d) For every language L, the language L^* is non-empty.

   (e) Every regular language is recognized by an NFA with exactly 374 accepting states.

   (f) If L does not have a fooling set of size 374, then L is regular.

   (g) The language $\left\{0^{374n} \mid n \geq 374\right\}$ is regular.

   (h) The language $\left\{0^{37n}1^{4n} \mid n \geq 374\right\}$ is regular.

   (i) The language $\left\{0^{3n}1^{74n} \mid n \leq 374\right\}$ is regular.

   (j) The empty language is context-free.

2. For any string $w \in \{0,1\}^*$, let $slash(w)$ be the string in $\{0,1,/\}^*$ obtained from w by inserting a new symbol / between any two consecutive symbols that are *not* equal. For example:

$$slash(\varepsilon) = \varepsilon$$
$$slash(00000) = 00000$$
$$slash(000110111) = 000/11/0/111$$

   For any language $L \subseteq \{0,1\}^*$, let $slash(L) = \{slash(w) \mid w \in L\}$.

   (a) Draw or describe a DFA that accepts the language $slash(\{0,1\}^*)$.

   (b) Give a regular expression for the language $slash(\{0,1\}^*)$.

   (c) ***Prove*** that for any regular language L, the language $slash(L)$ is also regular.

   (You do not need to justify your answers to parts (a) and (b).)

3. Let L be the language $\left\{0^a 1^b 0^c \;\middle|\; a + b = 2c\right\}$
   (a) ***Prove*** that L is not a regular language.
   (b) Describe a context-free grammar for L. (You do not need to justify your answer.)

4. For each of the following languages L, give a regular expression that represents L ***and*** draw or describe a DFA that recognizes L. You do not need to justify your answers.
   (a) All strings in $\{0, 1\}^*$ that do not contain either 001 or 110 as a substring
   (b) All strings in $\{0, 1, 2\}^*$ that do not contain either 01 or 12 as a substring

5. For any string $w \in \{0, 1\}^*$, let *obliviate*(w) denote the string obtained from w by removing every 1. For example:

$$
\begin{aligned}
\mathit{obliviate}(\varepsilon) &= \varepsilon\\
\mathit{obliviate}(000000) &= 000000\\
\mathit{obliviate}(111111) &= \varepsilon\\
\mathit{obliviate}(0100001101) &= 000000
\end{aligned}
$$

   Let L be an arbitrary regular language.
   (a) ***Prove*** that the language $\{\mathit{obliviate}(w) \mid w \in L\}$ is regular.
   (b) ***Prove*** that the language $\{w \in \{0, 1\}^* \mid \mathit{obliviate}(w) \in L\}$ is regular.

CS/ECE 374 A ✧ Fall 2019

Fake Midterm 2

November 11, 2019

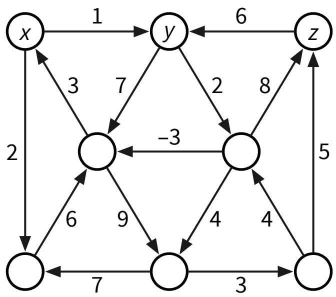
Real name:	
NetID:	

Gradescope name:	
Gradescope email:	

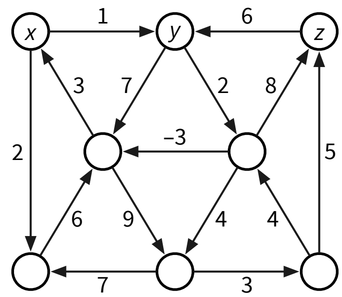
- ***Don't panic!***
- **All problems are described in more detail in a separate handout.** If any problem is unclear or ambiguous, please don't hesitate to ask us for clarification.
- If you brought anything except your writing implements, your **hand-written** double-sided 8½" × 11" cheat sheet, and your university ID, 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. However, if you are using your real name and your university email address on Gradescope, you do ***not*** need to write everything twice. **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.
- If you run out of space for an answer, feel free to use the scratch pages at the back of the answer booklet, but **please clearly indicate where we should look**.
- Except for greedy algorithms, proofs are required for full credit if and only if we explicitly ask for them, using the word ***prove*** in bold italics.
- Please return ***all*** paper with your answer booklet: your question sheet, your cheat sheet, and all scratch paper.

CS/ECE 374 A ✧ Fall 2019 Fake Midterm 2 Problem 1	Gradescope name:

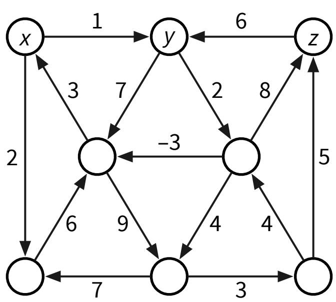
Clearly indicate the following structures in the directed graph below, or write NONE if the indicated structure does not exist. Don't be subtle; to indicate a collection of edges, draw a heavy black line along the entire length of each edge.



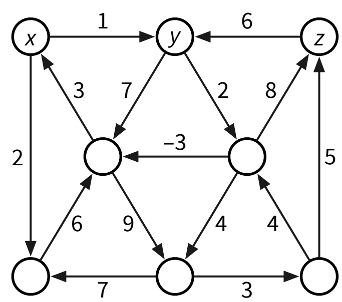
(a) A depth-first tree rooted at x.



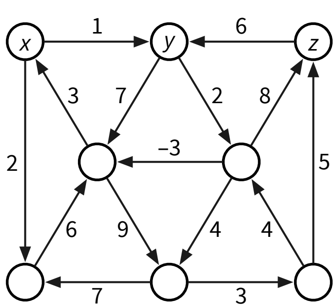
(b) A breadth-first tree rooted at y.



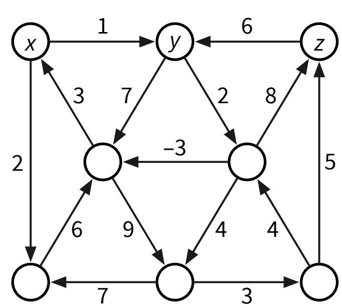
(c) A shortest-path tree rooted at z.



(d) The shortest directed cycle.



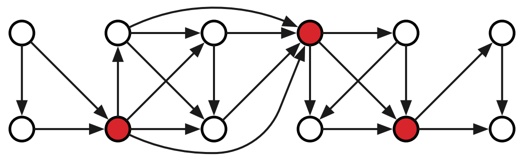
[scratch]



[scratch]

CS/ECE 374 A ✧ Fall 2019 Fake Midterm 2 Problem 2	Gradescope name:

A vertex v in a (weakly) connected graph G is called a ***cut vertex*** if the subgraph $G - v$ is disconnected. For example, the following graph has three cut vertices, which are shaded in the figure.



Suppose you are given a (weakly) connected *dag* G with one source and one sink. Describe and analyze an algorithm that returns TRUE if G has a cut vertex and FALSE otherwise.

CS/ECE 374 A ✧ Fall 2019 Fake Midterm 2 Problem 3	Gradescope name:

The City Council of Sham-Poobanana needs to partition Purple Street into voting districts. A total of n people live on Purple Street, at consecutive addresses $1, 2, \ldots, n$. Each voting district must be a contiguous interval of addresses $i, i+1, \ldots, j$ for some $1 \le i < j \le n$. By law, each Purple Street address must lie in exactly one district, and the number of addresses in each district must be between k and $2k$, where k is some positive integer parameter.

Every election in Sham-Poobanana is between two rival factions: Oceania and Eurasia. A majority of the City Council are from Oceania, so they consider a district to be *good* if more than half the residents of that district voted for Oceania in the previous election. Naturally, the City Council has complete voting records for all n residents.

For example, the figure below shows a legal partition of 22 addresses into 4 good districts and 3 bad districts, where $k = 2$. Each O indicates a vote for Oceania, and each X indicates a vote for Eurasia.

Describe an algorithm to find the largest possible number of *good* districts in a legal partition. Your input consists of the integer k and a boolean array GoodVote$[1..n]$ indicating which residents previously voted for Oceania (True) or Eurasia (False). You can assume that a legal partition exists. Analyze the running time of your algorithm in terms of the parameters n and k.

CS/ECE 374 A ✧ Fall 2019 Fake Midterm 2 Problem 4	Gradescope name:

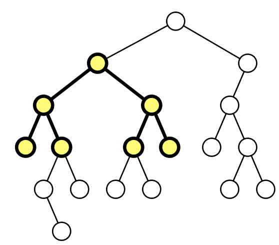
After graduation, you accept a job with Aviophiles-Я-Us, the leading traveling agency for people who love to fly. Your job is to build a system to help customers plan airplane trips from one city to another. Your customers love flying, but they absolutely despise airports. You know all the departure and arrival times of all the flights on the planet.

Suppose one of your customers wants to fly from city X to city Y. Describe an algorithm to find a sequence of flights that *minimizes the total time spent in airports*. Assume (unrealistically) that your customer can enter the starting airport immediately before the first flight leaves X, that they can leave the final airport at Y immediately after the final flight arrives at Y.

CS/ECE 374 A ✧ Fall 2019 Fake Midterm 2 Problem 5	Gradescope name:

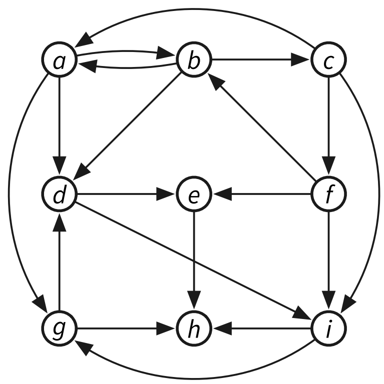
For this problem, a *subtree* of a binary tree means any connected subgraph. A binary tree is *complete* if every internal node has two children, and every leaf has exactly the same depth.

Describe and analyze a recursive algorithm to compute the ***largest complete subtree*** of a given binary tree. Your algorithm should return both the root and the depth of this subtree. For example, given the following tree T as input, your algorithm should return the left child of the root of T and the integer 2.



Write your answers in the separate answer booklet.
Please return this question sheet and your cheat sheet with your answers.

1. ***Clearly*** indicate the following structures in the directed graph below, or write NONE if the indicated structure does not exist. Don't be subtle; to indicate a collection of edges, draw a heavy black line along the entire length of each edge.



   (a) A depth-first search tree rooted at vertex a.
   (b) A breadth-first tree rooted at vertex a.
   (c) The strong components of G. (Circle each strong component.)
   (d) Draw the strong-component graph of G.

2. As the days get shorter in winter, Eggsy Hutmacher is increasingly worried about his walk home from work. The city has recently been invaded by the notorious Antimilliner gang, whose members hang out on dark street corners and steal hats from unwary passers-by, and a gentleman is simply *not* seen out in public without a hat. The city council is slowly installing street lamps at intersections to deter the Antimilliners, whose uncovered faces can be easily identified in the light. Eggsy keeps k extra hats in his briefcase in case of theft or other millinery emergencies.

   Eggsy has a map of the city in the form of an undirected graph G, whose vertices represent intersections and whose edges represent streets between them. A subset of the vertices are marked to indicate that the corresponding intersections are lit. Every edge e has a non-negative length $\ell(e)$. The graph has two special nodes s and t, which represent Eggsy's work and home, respectively.

   Describe an algorithm that computes the shortest path in G from s to t that visits at most k unlit vertices.

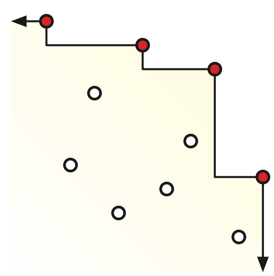
3. An undirected graph $G = (V, E)$ is ***bipartite*** if each of its vertices can be colored either black or white, so that every edge in E has one white endpoint and one black endpoint. Describe and analyze an algorithm to determine, given an undirected graph G as input, whether G is bipartite. *[Hint: Every tree is bipartite.]*

4. Satya is in charge of establishing a new testing center for the Standardized Awesomeness Test (SAT), and found an old conference hall that is perfect. The conference hall has n rooms of various sizes along a single long hallway, numbered in order from 1 through n. Satya knows exactly how many students fit into each room, and he wants to use a subset of the rooms to host as many students as possible for testing.

   Unfortunately, there have been several incidents of students cheating at other testing centers by tapping secret codes through walls. To prevent this type of cheating, Satya can use two adjacent rooms only if he demolishes the wall between them. The city's chief architect has determined that demolishing the walls on both sides of the same room would threaten the building's structural integrity. For this reason, Satya can never host students in three consecutive rooms.

   Describe an efficient algorithm that computes the largest number of students that Satya can host for testing without using three consecutive rooms. The input to your algorithm is an array $S[1..n]$, where each $S[i]$ is the (non-negative integer) number of students that can fit in room i.

5. Suppose you are given a set P of n points in the plane. A point $p \in P$ is *maximal* in P if no other point in P is both above and to the right of P. Intuitively, the maximal points define a "staircase" with all the other points of P below it.

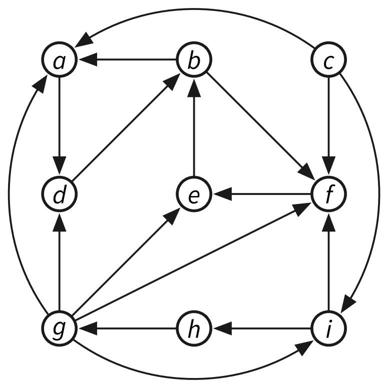


A set of ten points, four of which are maximal.

   Describe and analyze an algorithm to compute the number of maximal points in P in $O(n \log n)$ time. For example, given the ten points shown above, your algorithm should return the integer 4. The input to your algorithm is a pair of arrays $X[1..n]$ and $Y[1..n]$ containing the x- and y-coordinates of the points in P.

> **Write your answers in the separate answer booklet.**
> Please return this question sheet and your cheat sheet with your answers.

1. ***Clearly*** indicate the following structures in the directed graph below, or write NONE if the indicated structure does not exist. Don't be subtle; to indicate a collection of edges, draw a heavy black line along the entire length of each edge.



   (a) A depth-first search tree rooted at vertex c.

   (b) A breadth-first tree rooted at vertex c.

   (c) The strong components of G. (Circle each strong component.)

   (d) Draw the strong-component graph of G.

2. During her walk to work every morning, Rachel likes to buy a cappuccino at a local coffee shop, and a croissant at a local bakery. Her home town has *lots* of coffee shops and lots of bakeries, but strangely never in the same building. Punctuality is not Rachel's strongest trait, so to avoid losing her job, she wants to follow the shortest possible route.

   Rachel has a map of her home town in the form of an undirected graph G, whose vertices represent intersections and whose edges represent roads between them. A subset of the vertices are marked as bakeries; another disjoint subset of vertices are marked as coffee shops. The graph has two special nodes s and t, which represent Rachel's home and work, respectively.

   Describe an algorithm that computes the shortest path in G from s to t that visits both a bakery and a coffee shop, or correctly reports that no such path exists.

3. An undirected graph $G = (V, E)$ is ***bipartite*** if its vertices can be partitioned into two subsets L and R, such that every edge in E has one endpoint in L and one endpoint in R. Describe and analyze an algorithm to determine, given an undirected graph G as input, whether G is bipartite. *[Hint: Every tree is bipartite.]*

4. Satya is in charge of establishing a new testing center for the Standardized Awesomeness Test (SAT), and found an old conference hall that is perfect. The conference hall has n rooms of various sizes along a single long hallway, numbered in order from 1 through n. Each pair of adjacent rooms i and $i+1$ is separated by a single wall. Satya knows exactly how many students fit into each room, and he wants to use a subset of the rooms to host as many students as possible for testing.

   Unfortunately, there have been several incidents of students cheating at other testing centers by tapping secret codes through walls. To prevent this type of cheating, Satya can use two adjacent rooms only if he demolishes the wall between them. For example, if Satya wants to use rooms 1, 3, 4, 5, 7, 8, and 10, he must demolish three walls: between rooms 3 and 4, between rooms 4 and 5, and between rooms 7 and 8.

   The city's chief architect has determined that demolishing more than k walls would threaten the structural integrity of the building.

   Describe an efficient algorithm that computes the largest number of students that Satya can host for testing without demolishing more than k walls. The input to your algorithm is the integer k and an array $S[1..n]$, where each $S[i]$ is the (non-negative integer) number of students that can fit in room i.

5. Suppose you are given an array $A[1..n]$ of numbers.

   (a) Describe and analyze an algorithm that either returns two indices i and j such that $A[i]+A[j]=374$, or correctly reports that no such indices exist.

   (b) Describe and analyze an algorithm that either returns three indices i, j, and k such that $A[i]+A[j]+A[k]=374$, or correctly reports that no such indices exist.

   Do ***not*** use hashing. As always, faster correct algorithms are worth more points.

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 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 $L \subseteq \{0, 1\}^*$?

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 $\overline{L}$ is undecidable.

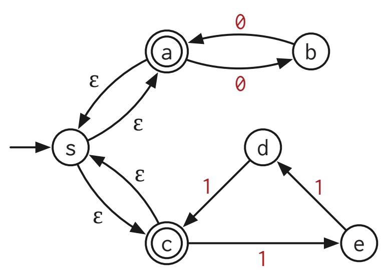
(b) Consider the following sets of undirected graphs:

- 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≠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	ε-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 $S \to 0S \mid 10S \mid \varepsilon$

(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.

$$\text{MARVIN} := \left\{ \langle M \rangle \;\middle|\; M \text{ rejects an infinite number of strings} \right\}$$

Professor Prefect, your instructor in Vogon Poetry and Knowing Where Your Towel Is, 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 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 ⟨HoopyFrood⟩.
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:

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.

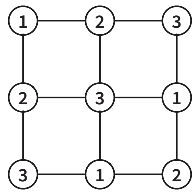
You have converted the trail map into a directed graph $G = (V, E)$, whose vertices represent trail intersections, and whose edges represent trail segments. A subset $R \subseteq V$ of the vertices indicate the locations of the *R*epellent *R*efilling stations, and a subset $B \subseteq E$ of the edges 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.

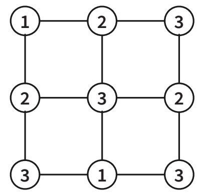
(b) Describe and analyze an algorithm to decide if you can walk safely from any refill station any other refill station. In other words, for *every* pair of vertices u and v in R, is there a safe path from u to v?

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.

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:

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

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

(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 Final Exam Problem 6	Gradescope name:

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 $S_1, S_2, \ldots, S_m$ 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?

CS/ECE 374 A ✧ Fall 2019

Final Exam

December 16, 2019

Real name:	
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.**
- **Print 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.**
- Good luck, and thanks for a great semester!

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 are true for ***at least one*** language $L \subseteq \{0, 1\}^*$?

Yes	No	IDK	L^* is empty.
Yes	No	IDK	L^* is not regular.
Yes	No	IDK	L^* is decidable.
Yes	No	IDK	L is decidable but L^* is undecidable.
Yes	No	IDK	L is the intersection of two regular languages, and L is undecidable.

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

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

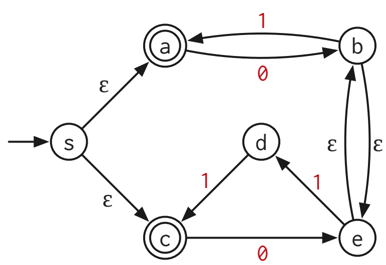
(c) Consider the following sets of undirected graphs:

- TREES is the set of all connected undirected graphs with no cycles.
- MOSTLYINDEPENDENT is the set of all undirected graphs that have an independent set containing at least half of the vertices. (Deciding whether a graph has this property is NP-hard.)

(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 $\notin$ P
Yes	No	IDK	TREES $\subseteq$ MOSTLYINDEPENDENT
Yes	No	IDK	There is a polynomial-time reduction from TREES to MOSTLYINDEPENDENT
Yes	No	IDK	There is a polynomial-time reduction from MOSTLYINDEPENDENT to TREES
Yes	No	IDK	TREES is NP-hard.

(d) Let M be the following NFA:



Which of the following statements about M are true?

Yes	No	IDK	M rejects the empty string ε.
Yes	No	IDK	$\delta^*(s, \texttt{0101}) = \{a, d\}$
Yes	No	IDK	ε-reach$(s) = \{s, a, c\}$
Yes	No	IDK	M accepts the string 01101011
Yes	No	IDK	$L(M) = (\texttt{011})^* + (\texttt{01})^*$

(e) 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	$\left\{w \in \Sigma^* \;\middle|\; \#(1, w) \text{ is a perfect square}\right\}$
Yes	No	IDK	The language generated by the context-free grammar $S \to 0S \mid S1 \mid \varepsilon$
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 complement of a regular language

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

Yes	No	IDK	3Color
Yes	No	IDK	$\{\langle M \rangle \mid M$ accepts all non-empty strings but rejects the empty string $\varepsilon\}$
Yes	No	IDK	$\{\langle M \rangle \mid M$ accepts every string whose length is a perfect square$\}$
Yes	No	IDK	$\{\langle M \rangle \mid M$ is a Turing machine with at least two states$\}$
Yes	No	IDK	$\varnothing$

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

Yes	No	IDK	3Color
Yes	No	IDK	$\{\langle M \rangle \mid M$ accepts all non-empty strings but rejects the empty string $\varepsilon\}$
Yes	No	IDK	$\{\langle M \rangle \mid M$ accepts every string whose length is a perfect square$\}$
Yes	No	IDK	$\{\langle M \rangle \mid M$ is a Turing machine with at least two states$\}$
Yes	No	IDK	$\varnothing$

(h) 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 Beeblebrox, your instructor in Infinitely Improbable Galactic Presidencies, 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 Beeblebrox claims that the following algorithm decides Halt.

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

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

Yes	No	IDK	If M accepts w, then HeartOfGold accepts VOGONPOETRY.
Yes	No	IDK	If M rejects w, then HeartOfGold rejects VOGONPOETRY.
Yes	No	IDK	If M hangs on w, then HeartOfGold rejects EDDIE.
Yes	No	IDK	ParanoidAndroid rejects ⟨HeartOfGold⟩.
Yes	No	IDK	DecideHalt decides Halt; that is, Professor Beeblebrox's proof is correct.

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

You and your friends are planning a hiking trip in Jellystone National Park over winter break. You have a map of the park's trails that lists all the scenic views in the park but also warns that certain trail segments have a high risk of bear encounters. To make the hike worthwhile, you want to see at least three scenic views. You also don't want to get eaten by a bear, so you are willing to hike at most one high-bear-risk segment. Because the trails are narrow, each trail segment allows traffic in only one direction.

Your friend has converted the map into a directed graph $G = (V, E)$, where V is the set of intersections and E is the set of trail segments. A subset S of the edges are marked as *Scenic*; another subset B of the edges are marked as *high-Bear-risk*. You may assume that $S \cap B = \varnothing$. Each segment $e \in E$ is also labeled with a positive length $\ell(e)$ in miles. Your campsite appears on the map as a particular vertex $s \in V$, and the visitor center is another vertex $t \in V$.

Describe and analyze an algorithm to compute the shortest hike from your campsite s to the visitor center t that includes *at least* three scenic views and *at most* one high-bear-risk trail segment. You may assume such a hike exists.

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

For each of the following languages over the alphabet $\{0, 1\}$, 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) $\{xy \mid x \text{ is a palindrome and } y \text{ is a palindrome}\}$

(b) $\{xy \mid x \text{ is a palindrome and } |x| \geq 2\}$

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

Vankin's Mile is an American solitaire game played on an $n \times n$ square grid. The player starts by placing a token on any square of the grid. Then on each turn, the player moves the token either one square to the right or one square down. The game ends when player moves the token off the edge of the board. Each square of the grid has a numerical value, which could be positive, negative, or zero. The player starts with a score of zero; whenever the token lands on a square, the player adds its value to his score. The object of the game is to score as many points as possible.

For example, given the grid shown below, the player can score $7-2+3+5+6-4+8+0 = 23$ points by following the path on the left, or they can score $8-4+1+5+1-4+8 = 15$ points by following the path on the right.

−1	**7**⇒	**−2**⇓	10	−5
8	−4	**3**⇓	−6	0
5	1	**5**⇒	**6**⇓	−5
−7	−4	1	**−4**⇒	**8**⇓
7	1	−9	4	**0**⇓

−1	7	−2	10	−5
8⇒	**−4**⇓	3	−6	0
5	**1**⇒	**5**⇓	6	−5
−7	−4	**1**⇒	**−4**⇒	**8**⇒
7	1	−9	4	0

(a) Describe and analyze an efficient algorithm to compute the maximum possible score for a game of Vankin's Mile, given the $n \times n$ array of values as input.

(b) A variant called *Vankin's Niknav* adds an additional constraint to Vankin's Mile: *The sequence of values that the token touches must be a* ***palindrome***. Thus, the example path on the right is valid, but the example path on the left is not. Describe and analyze an efficient algorithm to compute the maximum possible score for an instance of Vankin's Niknav, given the $n \times n$ array of values as input.

4 (continued)

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

Recall that a *satisfying assignment* for a 3CNF Boolean formula Φ assigns values (True or False) to the variables of Φ so that Φ evaluates to True. A satisfying assignment is *balanced* if *exactly* half of the variables are set to True.

The Balanced3Sat problem asks whether a given 3CNF formula Φ has a balanced satisfying assignment. ***Prove*** that Balanced3Sat is NP-hard.

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

Let M be an arbitrary NFA *without* ε-transitions, with input alphabet $\Sigma = \{0, 1\}$. Describe and analyze an efficient algorithm to decide whether M accepts 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 $S_1, S_2, \ldots, S_m$ 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?