CS/ECE 374 A ✧ Fall 2023

Practice Final Exam 2

December 7, 2023

Name:	
NetID:	

- ***Don't panic!***
- You have 180 minutes to answer six numbered questions. The questions are described in more detail in a separate handout.
- If you brought anything except your writing implements, your two 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 name and your NetID in the boxes above.
- Please also print your name at the top of every page of the answer booklet, except this cover page. We want to make sure that if a staple falls out, we can reassemble your answer booklet. (It doesn't happen often, but it does happen.)
- **Do not write outside the black boxes on each page**. These indicate the area of the page that our scanner will actually scan. If the scanner can't see your work, we can't grade it.
- If you run out of space for an answer, please use the overflow pages at the back of the answer booklet, but **please clearly indicate where we should look**. If we can't find your work, we can't grade it.
- We will provide scratch paper to anyone who asks, but **only work that is written into the stapled answer booklet will be graded**. In particular, we will not grade any pages that you separate from the answer booklet.
- Proofs or other justifications are required for full credit if and only if we explicitly ask for them, using the word ***prove*** or ***justify*** in bold italics.
- Breathe in. Breathe out. You've got this.

CS/ECE 374 A ✧ Fall 2023 Practice Final 2 Problem 1	Name:

Recall that a ***run*** in a string $w \in \{0,1\}^*$ is a maximal substring of w whose characters are all equal.

(a) Let L_a denote the set of all non-empty strings in $\{0,1\}^*$ where the length of the first run is equal to the number of runs. ***Prove*** that L_a is not a regular language.

(b) Let L_b denote the set of all strings in $\{0,1\}^*$ that contain an even number of odd-length runs. Describe a DFA or NFA that accepts L_b ***and*** give a regular expression that describes L_b. (You do not need to prove that your answers are correct.)

CS/ECE 374 A ✧ Fall 2023 Practice Final 2 Problem 2	Name:

Aladdin and Badroulbadour are playing a cooperative game. Each player has an array of positive integers, arranged in a row of squares from left to right. Each player has a token, which starts at the leftmost square of their row; their goal is to move *both* tokens onto the rightmost squares at the same time.

On each turn, *both* players move their tokens *in the same direction*, either left or right. The distance each token travels is equal to the number under that token at the beginning of the turn. If *either* token moves past either end of its row, then both players immediately lose.

Describe and analyze an algorithm to determine whether Aladdin and Badroulbadour can solve their puzzle, given the input arrays $A[1..n]$ and $B[1..n]$.

CS/ECE 374 A ✧ Fall 2023 **Practice Final 2 Problem 3**	Name:

Submit a solution to ***exactly one*** of the following problems. Don't forget to tell us which problem you've chosen!

(a) Let $G = (V, E)$ be an arbitrary undirected graph. A subset $S \subseteq V$ of vertices is *mostly independent* if more than half the vertices of S have no neighbors in S. ***Prove*** that finding the largest mostly independent set in G is NP-hard.

(b) ***Prove*** that the following problem is NP-hard: Given an undirected graph G, find the largest integer k such that G contains *two disjoint* independent sets of size k.

(In fact, both of these problems are NP-hard, but we only want a proof for one of them.)

CS/ECE 374 A ✧ Fall 2023 Practice Final 2 Problem 4	Name:

(a) Describe and analyze 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. Describe and analyze 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).]*

CS/ECE 374 A ✧ Fall 2023 Practice Final 2 Problem 5	Name:

You have a collection of n lockboxes and m gold keys. Each key unlocks *at most* one box. Without a matching key, the only way to open a box is to smash it with a hammer. Your baby brother has locked all your keys inside the boxes! Luckily, you know which keys (if any) are inside each box.

(a) Your baby brother has found the hammer and is eagerly eyeing one of the boxes. Describe and analyze an algorithm to determine if it is possible to retrieve all the keys without smashing any box except the one your brother has chosen.

(b) Describe and analyze an algorithm to compute the minimum number of boxes that must be smashed to retrieve all the keys.

CS/ECE 374 A ✧ Fall 2023 Practice Final 2 Problem 6	Name:

For each statement below, there are two boxes in the answer booklet labeled "Yes" and "No". Check "Yes" if the statement is ***always*** true and "No" otherwise, and give a ***brief*** (at most one short sentence) explanation of your answer. **Assume $P \neq NP$.** If there is any other ambiguity or uncertainty about an answer, check "No". Read each statement *very* carefully; some of these are deliberately subtle!

(a) Which of the following statements are true for ***all*** languages $L \subseteq \{0,1\}^*$?

- $L^* = (L^*)^*$

  Yes No _____

- If L is decidable, then L^* is decidable.

  Yes No _____

- L is either regular or NP-hard.

  Yes No _____

- If L is undecidable, then L has an infinite fooling set.

  Yes No _____

- The language $\{\langle M \rangle \mid M \text{ decides } L\}$ is undecidable.

  Yes No _____

Problem 6 continues onto the next three pages.

(b) Which of the following statements are true?

- The solution to the recurrence $T(n) = 4T(n/4) + O(n)$ is $T(n) = O(n \log n)$.

  Yes No _____

- The solution to the recurrence $T(n) = 4T(n/4) + O(n^2)$ is $T(n) = O(n^2 \log n)$.

  Yes No _____

- Every directed acyclic graph contains at most one source and at most one sink.

  Yes No _____

- Depth-first search explores every path from the source vertex s to every other vertex in the input graph.

  Yes No _____

- Suppose $A[1..n]$ is an array of integers. Consider the following recursive function:

$$Huh(i,j) = \begin{cases} 0 & \text{if } i < 0 \text{ or } j > n \\ \max \left\{ \begin{array}{c} Huh(i, j+1) \\ Huh(i-1, j) \\ A[i] \cdot A[j] + Huh(i-1, j+1) \end{array} \right\} & \text{otherwise} \end{cases}$$

  We can compute $Huh(n, 0)$ by memoizing this function into an array $Huh[0..n, 0..n]$ in $O(n^2)$ time, increasing i in the outer loop and increasing j in the inner loop.

  Yes No _____

Problem 6 continues onto the next two pages.

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

$$\textsc{Muggle} := \{\langle M \rangle \mid M \text{ accepts } \texttt{SCIENCE} \text{ but rejects } \texttt{MAGIC}\}$$

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

$$\textsc{Halt} := \{(\langle M \rangle, w) \mid M \text{ halts on input } w\}.$$

Specifically, suppose there is a Turing machine DetectoMuggletum that decides Muggle. Professor Potter claims that the following algorithm decides Halt.

```
DecideHalt(⟨M⟩, w):
  Write code for the following algorithm:
      RubberDuck(x):
        run M on input w
        ⟨⟨ignore the output of M⟩⟩
        if x = MAGIC
            return False
        else
            return True
  return DetectoMuggletum(⟨RubberDuck⟩)
```

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

- If M accepts w, then RubberDuck accepts MAGIC.

  Yes No _____

- If M diverges on w, then RubberDuck rejects MAGIC.

  Yes No _____

- If M accepts w, then DetectoMuggletum accepts $\langle$RubberDuck$\rangle$.

  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 _____

Problem 6 continues onto the next page.

(d) Suppose there is a ***polynomial-time*** reduction from some language $A \subseteq \{0,1\}$ reduces to some other language $B \subseteq \{0,1\}$. Which of the following statements are true, assuming P $\neq$ NP?

- $A \cap B \neq \emptyset$.

  Yes No _____

- There is an algorithm to transform any Python program that solves B in polynomial time into a Python program that solves A in polynomial time.

  Yes No _____

- If B is NP-hard, then A is NP-hard.

  Yes No _____

- If B is decidable, then A is decidable.

  Yes No _____

- If a Turing machine M accepts every string in B, the *same* Turing machine M also accepts every string in A.

  Yes No _____

(overflow / scratch paper)

(overflow / scratch paper)

(overflow / scratch paper)

(overflow / 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?

ChromaticNumber: Given an undirected graph G, what is the minimum number of colors required to color its vertices, 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?