CS/ECE 374 A ✧ Spring 2026

Practice Final Exam 2

May 6, 2026

Name:	
NetID:	

- ***Don't panic!***
- You have 180 minutes to answer seven 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 sheets, 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.)
- Greedy algorithms require formal proofs of correctness to receive any credit, even if they are correct. Otherwise, proofs or other justifications beyond items listed in the standard rubrics are required for full credit if and only if we explicitly ask for them, using the word ***prove*** or ***justify*** in bold italics. (*Problem 5 part (b)* contains a specific exception to this instruction.)

- **Please do not write outside the black boxes on each page.** These indicate the area of the page that our scanners can 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/scratch 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.
- **Only work that is written into the stapled answer booklet will be graded**. In particular, you are welcome to detach scratch pages from the answer booklet, but any work on those detached pages will not be graded. We will provide additional scratch paper on request, but any work on that scratch paper will not be graded.
- Please return ***all*** paper with your answer booklet: your question sheet, your cheat sheet, and all scratch paper. **Please put all loose paper *inside* your answer booklet.**
- Breathe in. Breathe out. You've got this.

This sentence contains four and three fourths percent a's, five and one half percent c's, three percent d's, eighteen and one fourth percent e's, four and one half percent f's, three fourths percent g's, five percent h's, two and one half percent i's, two percent l's, twelve and one half percent n's, six percent o's, five percent p's, eight percent r's, seven percent s's, nine and three fourths percent t's, two and one fourth percent u's, one and one half percent v's, one and one fourth percent w's and one half percent x's.

CS/ECE 374 A ✧ Spring 2026 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 ✧ Spring 2026 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 ✧ Spring 2026 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 the following problem is NP-*complete*: Given an undirected graph G and an integer k, decide if G contains a mostly independent set of size at least k.

(b) ***Prove*** that the following problem is NP-*complete*: Given an undirected graph G and an integer k, decide if G contains two disjoint independent sets of size k.

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

CS/ECE 374 A ✧ Spring 2026 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 ✧ Spring 2026 Practice Final 2 Problem 5	Name:

See the question sheets for a full description of the problem and its greedy solution.

1. ***Prove*** the greedy strategy described in the question sheets leads to a correct algorithm for the problem.

   [Hint: Let X^* be an optimal solution to the problem. Do an exchange argument to show there is an equally large solution X that contains ℓ.]

2. Describe an algorithm to find a maximum size set of numbers X as described in the question sheets. An algorithm based on the strategy in part (a) requires no further justification to receive full credit.

CS/ECE 374 A ✧ Spring 2026 Practice Final 2 Problem 6	Name:

For each statement below, 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 ≠ 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?

- 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 page.

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

$$\text{Duck} := \left\{ \langle M \rangle \;\middle|\; M \text{ accepts } \texttt{GRAPES} \text{ but rejects } \texttt{LEMONADE} \right\}$$

Professor Canard, your wetlands-ornithology instructor, suggests a reduction from the standard halting language

$$\text{Halt} := \left\{ (\langle M \rangle, w) \;\middle|\; M \text{ halts on input } w \right\}.$$

Specifically, suppose there is a Turing machine LooksLikeADuck that decides Duck. Professor Canard claims that the following algorithm decides Halt.

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

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

- If M accepts w, then WaddleAway accepts GRAPES.

  Yes | No | ________________

- If M diverges on w, then WaddleAway rejects GRAPES.

  Yes | No | ________________

- If M accepts w, then LooksLikeADuck accepts ⟨WaddleAway⟩.

  Yes | No | ________________

- If M diverges on w, then DecideHalt rejects $(\langle M \rangle, w)$.

  Yes | No | ________________

- DecideHalt decides the language Halt. (That is, Professor Canard's reduction is correct.)

  Yes | No | ________________

CS/ECE 374 A ✧ Spring 2026 Practice Final 2 Problem 7	Name:

For each statement below, 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 ≠ 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 ____

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

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

  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.

3Sat: Given a boolean formula in conjunctive normal form, with exactly three distinct literals per clause, does the formula have a satisfying assignment?

IndependentSet: Given an undirected graph G and an integer k, is there a subset of k vertices in G that have no edges among them?

Clique: Given an undirected graph G and an integer k, is there a subset of k vertices in G where there is an edge between every pair of them?

VertexCover: Given an undirected graph G and an integer k, is there a subset of k vertices that touch every edge in G?

DominatingSet: Given an undirected graph G and an integer k, is there a subset of k vertices such that every vertex of G is either in the set or adjacent to a member of the set?

SetCover: Given a collection of subsets $S_1, S_2, \ldots, S_m$ of a set S and an integer k, is there a subcollection of k of these subsets whose union is S?

HittingSet: Given a collection of subsets $S_1, S_2, \ldots, S_m$ of a set S and an integer k, is there a subset of S of size k 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 and a number L, is there a Hamiltonian path/cycle of weight at most L in G?

LongestPath: Given a graph G (either directed or undirected, possibly with weighted edges) and a number L, is there a simple path of length at least L in G?

SteinerTree: Given an undirected graph G with some of the vertices marked and an integer k, is there a subtree of G with at most k edges 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?

Draughts: Given an $n \times n$ international draughts configuration and an integer k, is there a move that can (and therefore must) capture at least k pieces in a single move?