CS/ECE 374 A ✧ Fall 2021

Conflict Final Exam

Directions

- ***Don't panic!***
- If you brought anything except your writing implements, your two hand-written double-sided $8\frac{1}{2}" \times 11"$ cheat sheets, please put it away for the duration of the exam. In particular, please turn off and put away *all* medically unnecessary electronic devices.
- **We *strongly* recommend reading the entire exam before trying to solve anything.** If you think a question is unclear or ambiguous, please ask for clarification as soon as possible.
- The exam has six numbered questions, each worth 10 points. (Subproblems are not necessarily worth the same number of points.)
- You have **150 minutes** to write your solutions, after which you have 30 minutes to scan your solutions, convert your scan to a PDF file, and upload your PDF file to Gradescope. (Both of these times are extended if you have time accommodations through DRES.)
- Proofs are required for full credit if and only if we explicitly ask for them, using the word ***prove*** in bold italics.
- Write your answers on blank white paper using a dark pen. Please start your solution to each numbered question on a new sheet of paper.
- If you are ready to scan your solutions and there are more than 15 minutes of writing time, send a private message to the host of your Zoom call ("Ready to scan") and wait for confirmation before leaving the Zoom call.
- Gradescope will only accept PDF submissions. Please do not scan your cheat sheets or scratch paper. Please make sure your solution to each numbered problem starts on a new page of your PDF file.
- Finally, if something goes seriously wrong, send email to jeffe@illinois.edu as soon as possible explaining the situation. If you have already finished the exam but cannot submit to Gradescope for some reason, include a complete scan of your exam **as a PDF file** in your email. If you are in the middle of the exam, send Jeff email, continue working until the time limit, and then send a second email with your completed exam **as a PDF file**. Please do not email raw photos.

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?

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?

1. For each statement below, write "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, write "NO". For example:

   - $x + y = 5$
     NO — Suppose $x = 3$ and $y = 4$.
   - 3SAT can be solved in polynomial time.
     NO — 3SAT is NP-hard.
   - If P = NP then Jeff is the Queen of England.
     YES — The hypothesis is false, so the implication is true.

   Read each statement *very* carefully; some of these are deliberately subtle!

Which of the following statements are true?

(a) The solution to the recurrence $T(n) = \mathbf{2}T(n/\mathbf{4}) + O(n^2)$ is $T(n) = O(n^2)$.

(b) The solution to the recurrence $T(n) = \mathbf{4}T(n/\mathbf{2}) + O(n^2)$ is $T(n) = O(n^2)$.

(c) For every directed graph G, if G has at least one source, then G has at least one sink.

(d) Given *any* undirected graph G, we can compute a spanning tree of G in $O(V + E)$ time using whatever-first search.

(e) Suppose we want to iteratively evaluate the following recurrence:

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

We can fill the array $What[0..n, 0..n]$ in $O(n^2)$ time, by decreasing i in the outer loop and decreasing j in the inner loop.

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

(f) $L^* = (L^*)^*$

(g) If L is decidable, then L^* is decidable.

(h) L is either regular or NP-hard.

(i) If L is undecidable, then L has an infinite fooling set.

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

---

2. For each statement below, write "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, write "NO".

   Read each statement *very* carefully; some of these are deliberately tricky!

   (Please remember to start your answers to this problem on a new page. Yes, this is really just a continuation of problem 1; we split it into two problems to make grading easier.)

Consider the following pair of languages:

- DIRHAMPATH := $\{G \mid G$ is a directed graph with a Hamiltonian path$\}$
- ACYCLIC := $\{G \mid G$ is a directed acyclic graph$\}$

(For concreteness, assume that in both of these languages, graphs are represented by their adjacency matrices.) Which of the following statements are true, assuming P ≠ NP?

(a) ACYCLIC ∈ NP

(b) ACYCLIC ∩ DIRHAMPATH ∈ P

(c) DIRHAMPATH is decidable.

(d) A polynomial-time reduction from DIRHAMPATH to ACYCLIC would imply P=NP.

(e) A polynomial-time reduction from ACYCLIC to DIRHAMPATH would imply P=NP.

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 ≠ NP?

(f) $A \subseteq B$.

(g) 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.

(h) If A is NP-hard then B is NP-hard.

(i) If A is decidable then B is decidable.

(j) If a Turing machine M accepts B, the same Turing machine M also accepts A.

---

3. 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 to the rightmost squares.

   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. For example, if a token starts on a square labeled 5, then it moves either five squares to the right or five squares to the left. If *either* token moves past either end of its row, then both players immediately lose.

   For example, if Aladdin and Badroulbadour are given the arrays

A:	7	5	4	1	2	3	3	2	3	1	4	2
B:	5	1	2	4	7	3	5	2	4	6	3	1

   they can win the game by moving right, left, left, right, right, left, right. On the other hand, if they are given the arrays

A:	2	3	5	1	3
B:	3	4	1	2	1

   they cannot win the game. (The first move must be to the right; then Aladdin's token moves out of bounds on the second turn.)

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

4. 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 less than half the vertices of S have a neighbor that is also in S. ***Prove*** that finding the largest mostly independent set in G is NP-hard.

   (b) Let $G = (V, E)$ be an arbitrary directed graph with colored edges. A *rainbow Hamiltonian cycle* in G is a cycle that visits every vertex of G exactly one, in which no pair of consecutive edges have the same color. ***Prove*** that it is NP-hard to decide whether G has a rainbow Hamiltonian cycle.

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

5. Suppose we are given an n-digit integer X. Repeatedly remove one digit from either end of X (your choice) until no digits are left. The *square-depth* of X is the maximum number of perfect squares that you can see during this process. For example, the number 32492 has square-depth 3, by the following sequence of removals:

$$32492 \to \overset{57^2}{3249} \to \overset{18^2}{324} \to 24 \to \overset{2^2}{4} \to \varepsilon.$$

Describe and analyze an algorithm to compute the square-depth of a given integer X, represented as an array $X[1..n]$ of n decimal digits. Assume you have access to a subroutine IsSquare that determines whether a given k-digit number (represented by an array of digits) is a perfect square ***in $O(k^2)$ time***.

6. Recall that a ***run*** in a string $w \in \{0, 1\}^*$ is a maximal substring of w whose characters are all equal. For example, the string 00011111110000 is the concatenation of three runs:

$$00011111110000 = 000 \bullet 1111111 \bullet 0000$$

(a) Let L_a denote the set of all strings in $\{0, 1\}^*$ in which every run of 1s has even length and every run of 0s has odd length.

- Describe a DFA or NFA that accepts L_a ***and***
- Give a regular expression that describes L_a.

(You do not need to prove that your answers are correct.)

(b) Let L_b denote the set of all strings in $\{0, 1\}^*$ in which every run of 0s is immediately followed by a *longer* run of 1s. ***Prove*** that L_b is *not* a regular language.

Both of these languages contain the strings 0111100011 and 110001111 and 111111 and the empty string ε, but neither language contains 000111 or 100011 or 0000.