CS/ECE 374 A ✧ Fall 2021

Final Exam

December 15, 2021

Directions

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

- If you brought anything except your writing implements, your two hand-written 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.

- **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 remaining, 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* ≠ *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 $T(n) = \mathbf{4}T(n/\mathbf{2}) + O(n^2)$ is $T(n) = O(n^2)$. $n^2 \lg n$ False

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

(c) Every directed acyclic graph contains at least one sink. True.

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

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

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

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

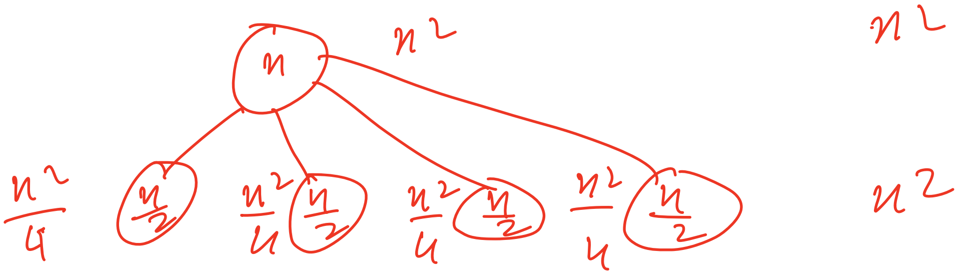
(f) $L^* = (L^*)^*$ True.

(g) L is decidable, but L^* is undecidable. False-

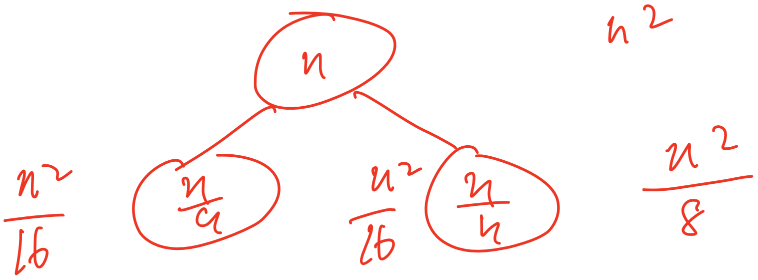
(h) L is neither regular nor NP-hard. True $L = \{0^n 1^n \mid n \geq 0\}$.

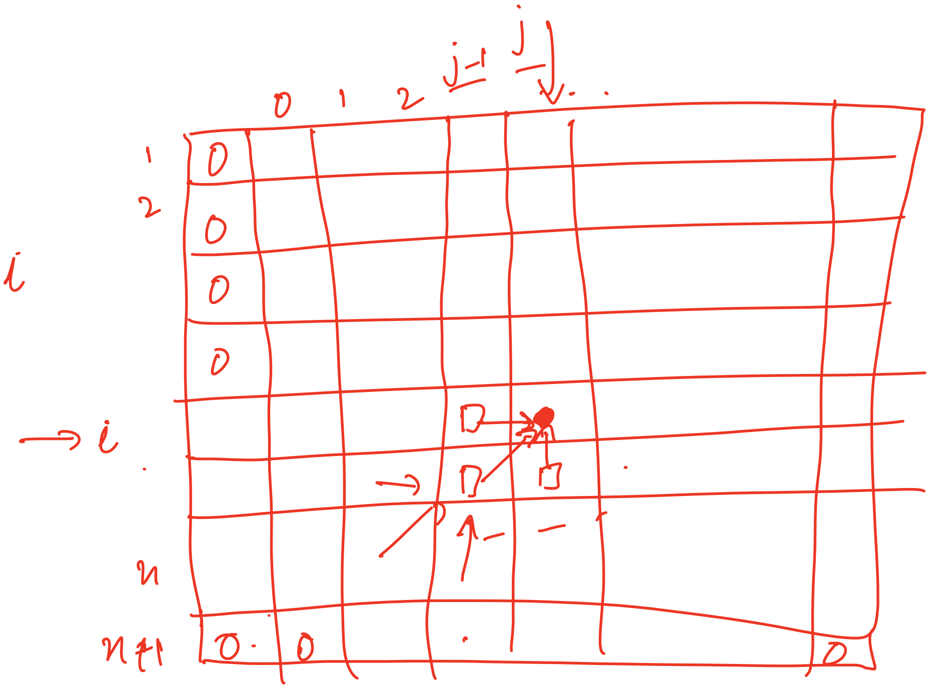
(i) L is in P, and L has an infinite fooling set.

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


n
n^2
n^2
n^2/4
n/2
n^2/4
n/2
n^2/4
n/2
n^2/4
n/2
n^2

$\boxed{\log_n}$


n
n^2
n^2/16
n/4
n^2/16
n/4
n^2/8


0
1
2
j-1
j
1
2
i
i
n
n+1
0
0
0
0
0
0
0

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 $P \neq NP$.** 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:

- ACYCLIC := $\{\text{undirected graph } G \mid G \text{ contains no cycles}\}$
- HALFIND := $\{\text{undirected graph } G = (V, E) \mid G \text{ has an independent set of size } |V|/2\}$

(For concreteness, assume that in both of these languages, graphs are represented by their adjacency matrices.) The language HALFIND is actually NP-hard; **you do *not* need to prove that fact**.

Which of the following statements are true, assuming P $\neq$ NP?

(a) ACYCLIC is NP-hard. False

(b) HALFIND \ ACYCLIC $\in$ P False
(Recall that $X \setminus Y$ is the subset of elements of X that are not in Y.)

(c) HALFIND is decidable. True.

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

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

---

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

$\boxed{A \le_p B}$

(f) A is a subset of B. False

(g) If $B \in P$, then $A \in P$. True

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

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

→ (j) If B is regular, then A is decidable. True.

(j') If B is regular then A is in P True

(j'') If B is regular then A is regular. False

Suppose L_1 is NP-Hard

L_2 is in P

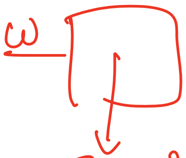
Is $L_1 - L_2$ NP-Hard or in P?

Prove that $\boxed{L_1 \leq_p L_1 - L_2}$

$\boxed{L = \{0^n 1^n \mid n \geq 0\} \quad B}$

$A = \{\underline{\underline{0, 1}}\}$.

$\underline{\underline{B \leq_p A}}$



If $w = 0^n 1^n$ for some n — output 0
else output 00.

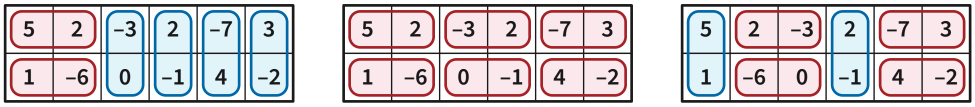
3. Suppose you are asked to tile a $2 \times n$ grid of squares with dominos (1×2 rectangles). Each domino must cover exactly two grid squares, either horizontally or vertically, and each grid square must be covered by exactly one domino.

   Each grid square is worth some number of points, which could be positive, negative, or zero. The ***value*** of a domino tiling is the sum of the points in squares covered by vertical dominos, *minus* the sum of the points in squares covered by horizontal dominos.

   Describe and analyze an efficient algorithm to compute the largest possible value of a domino tiling of a given $2 \times n$ grid. Your input is an array $Points[1..2, 1..n]$ of point values.

   As an example, here are three domino tilings of the same 2×6 grid, along with their values. The third tiling is optimal; no other tiling of this grid has larger value. Thus, given this 2×6 grid as input, your algorithm should return the integer 16.

5	2	−3	2	−7	3
1	−6	0	−1	4	−2



value = −6

value = 2

value = 16

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

   (a) Let Φ be a boolean formula in conjunctive normal form, with exactly three literals per clause (or in other words, an instance of 3SAT). ***Prove*** that it is NP-hard to decide whether Φ has a satisfying assignment in which *exactly half* of the variables are TRUE.

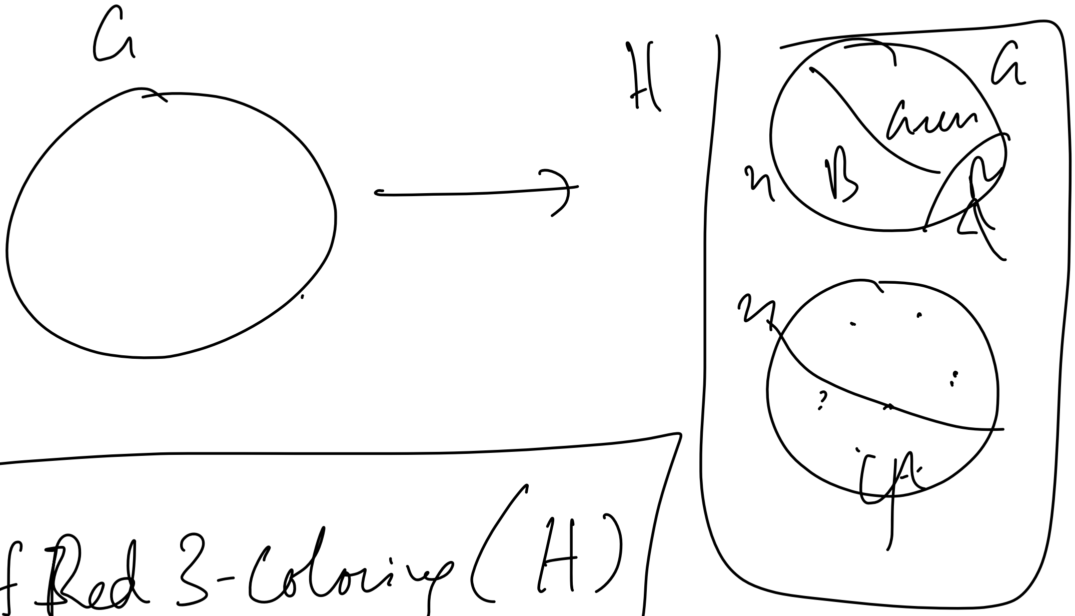
   (b) Let $G = (V, E)$ be an arbitrary undirected graph. Recall that a *proper 3-coloring* of G assigns each vertex of G one of three colors—red, blue, or green—so that every edge in G has endpoints with different colors. ***Prove*** that it is NP-hard to decide whether G has a proper 3-coloring in which *exactly half* of the vertices are red.

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

5. Suppose you are given a height map of a mountain, in the form of an $n \times n$ grid of evenly spaced points, each labeled with an elevation value. You can safely hike directly from any point to any neighbor immediately north, south, east, or west, but only if the elevations of those two points differ by at most Δ. (The value of Δ depends on your hiking experience and your physical condition.)

   Describe and analyze an algorithm to determine the longest hike from some point s to some other point t, where the hike consists of an uphill climb (where elevations must increase at each step) followed by a downhill climb (where elevations must decrease at each step). Your input consists of an array $Elevation[1..n, 1..n]$ of elevation values, the starting point s, the target point t, and the parameter Δ.

Hint: Graph modeling + Longest path in a DAG.



Half Red 3-Coloring(H)

MaxValue(i) : max value one can get by valid tiling of $2 \times i$ rectangle of the given input

$i = 0$ 0

$i = 1$?   Points[1,1] + Points[2,1]

$$2 \quad \max \begin{cases} \text{MaxValue}(i-1) + \text{Points}[1,i] + \text{Points}[2,i] \\ \text{MaxValue}(i-2) - \left(\text{Points}[1,i] + \text{Points}[2,i] + \text{Points}[1,i-1] + \text{Points}[2,i-1] \right) \end{cases}$$

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:

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

Should be OK for all of you.

(a) Let L_a denote the set of all strings in $\{0,1\}^*$ where every 0 is followed immediately by at least one 1.

For example, L_a contains the strings 010111 and 1111 and the empty string ε, but does not contain either 001100 or 1111110.

- 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 $\{0,1\}^*$ whose run lengths are increasing; that is, every run except the last is followed immediately by a *longer* run.

For example, L_b contains the strings 0110001111 and 1100000 and 000 and the empty string ε, but does not contain either 000111 or 100011.

Prove that L_b is not a regular language.

This topic not included in our final.