CS/ECE 374: Algorithms and Models of Computation, Fall 2016
Final Exam (Version X) — December 15, 2016

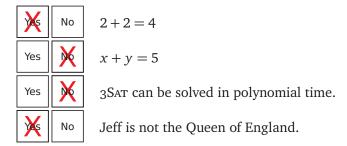| Name:	
NetID:	

Section:	A	B	C	D	E	F	G	H	I	J	K
	9–10	10–11	11–12	12–1	1–2	1–2	2–3	2–3	don't	3–4	3–4
	Spencer	Yipu	Charlie	Chao	Alex	Tana	Mark	Konstantinos	know	Mark	Konstantinos

#	1	2	3	4	5	6	Total
Score							
Max	20	10	10	10	10	10	70
Grader							

Assignment Project Exam Help

https://tutorcs.com

WeChat: cstutorcs

- *Don't panic!*

- Please print your name and your NetID and circle your discussion section in the boxes above.

- This is a closed-book, closed-notes, closed-electronics exam. 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, you may not use *any* electronic devices.

- **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".

- **Beware the Three Deadly Sins.** Give complete solutions, not examples. Don't use weak induction. Declare all your variables.

- If you use a greedy algorithm, you *must* prove that it is correct to receive any credit. Otherwise, proofs are required only when we explicitly ask for them.

- **Please return your cheat sheets and all scratch paper with your answer booklet.**

- *Good luck!* And have a great winter break!

1. 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	$2 + 2 = 4$
☐ Yes	☒ No	$x + y = 5$
☐ Yes	☒ No	3SAT can be solved in polynomial time.
☒ Yes	☐ No	Jeff is not the Queen of England.

There are 40 yes/no choices altogether.

- Each correct choice is worth $+\frac{1}{2}$ point.
- Each incorrect choice is worth $-\frac{1}{4}$ point.
- To indicate "I don't know", write **IDK** next to the boxes; each **IDK** is worth $+\frac{1}{8}$ point.

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

☐ Yes	☐ No	L contains the empty string ε.
☐ Yes	☐ No	L^* is infinite.
☐ Yes	☐ No	L^* is regular.
☐ Yes	☐ No	L is infinite or L is decidable (or both).
☐ Yes	☐ No	If L is the union of two regular languages, then L is regular.
☐ Yes	☐ No	If L is the union of two decidable languages, then L is decidable.
☐ Yes	☐ No	If L is the union of two undecidable languages, then L is undecidable.
☐ Yes	☐ No	L is accepted by some NFA with 374 states if and only if L is accepted by some DFA with 374 states.
☐ Yes	☐ No	If $L \notin \mathrm{P}$, then L is not regular.

(b) Which of the following languages over the alphabet $\{0, 1\}$ are **regular**?

Yes	No	$\{0^m 1^n \mid m \geq 0 \text{ and } n \geq 0\}$
Yes	No	All strings with the same number of 0s and 1s
Yes	No	Binary representations of all prime numbers less than 10^{100}
Yes	No	$\{ww \mid w \text{ is a palindrome}\}$
Yes	No	$\{wxw \mid w \text{ is a palindrome and } x \in \{0, 1\}^*\}$
Yes	No	$\{\langle M \rangle \mid M \text{ accepts a finite number of non-palindromes}\}$

(c) Which of the following languages over the alphabet $\{0, 1\}$ are **decidable**?

Yes	No	$\varnothing$
Yes	No	$\{0^n 1^{2n} 0^{3n} \mid n \geq 0\}$
Yes	No	$\{ww \mid w \text{ is a palindrome}\}$
Yes	No	$\{\langle M \rangle \mid M \text{ accepts a finite number of non-palindromes}\}$
Yes	No	$\{\langle M, w \rangle \mid M \text{ accepts } ww\}$
Yes	No	$\{\langle M, w \rangle \mid M \text{ accepts } ww \text{ after at most } |w|^2 \text{ transitions}\}$

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

Yes	No	$\{\langle M \rangle \mid M \text{ accepts an infinite number of strings}\}$
Yes	No	$\{\langle M \rangle \mid M \text{ accepts either } \langle M \rangle \text{ or } \langle M \rangle^R\}$
Yes	No	$\{\langle M \rangle \mid M \text{ does not accept exactly 374 palindromes}\}$
Yes	No	$\{\langle M \rangle \mid M \text{ accepts some string } w \text{ after at most } |w|^2 \text{ transitions}\}$

(e) Which of the following is a good English specification of a recursive function that can be used to compute the edit distance between two strings $A[1..n]$ and $B[1..n]$?

Yes	No	
Yes	No	$Edit(i, j)$ is the answer for i and j.
Yes	No	$Edit(i, j)$ is the edit distance between $A[i]$ and $B[j]$.
Yes	No	$Edit[1..n, 1..n]$ stores the edit distances for all prefixes.
Yes	No	$Edit(i, j)$ is the edit distance between $A[i..n]$ and $B[j..n]$.
Yes	No	$Edit[i, j]$ is the value stored at row i and column j of the table.

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

$$\textsc{Muggle} := \left\{ \langle M \rangle \;\middle|\; M \text{ accepts } \texttt{SCIENCE} \text{ but rejects } \texttt{MAGIC} \right\}$$

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

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

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

```
DecideHalt(⟨M, w⟩):
    Encode the following Turing machine:
        RubberDuck(x):
            run M on input w
            if x = MAGIC
                return False
            else
                return True
    return DetectoMuggletum(⟨RubberDuck⟩)
```

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

Yes	No	
Yes	No	If M rejects w, then RubberDuck rejects MAGIC.
Yes	No	If M accepts w, then DetectoMuggletum accepts $\langle$RubberDuck$\rangle$.
Yes	No	If M rejects w, then DecideHalt rejects $\langle M, w \rangle$.
Yes	No	DecideHalt decides the language Halt. (That is, Professor Potter's reduction is actually correct.)
Yes	No	DecideHalt actually runs (or simulates) RubberDuck.

(g) Consider the following pair of languages:

- HamPath := $\{G \mid G$ 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 *must* be true, assuming P≠NP?

Yes	No	Acyclic ∈ NP
Yes	No	Acyclic ∩ HamPath ∈ P
Yes	No	HamPath is decidable.
Yes	No	There is no polynomial-time reduction from HamPath to Acyclic.
Yes	No	There is no polynomial-time reduction from Acyclic to HamPath.

Assignment Project Exam Help

https://tutorcs.com

WeChat: cstutorcs

2. A **quasi-satisfying assignment** for a 3CNF boolean formula Φ is an assignment of truth values to the variables such that *at most one* clause in Φ does not contain a true literal. **Prove** that it is NP-hard to determine whether a given 3CNF boolean formula has a quasi-satisfying assignment.

Assignment Project Exam Help

https://tutorcs.com

WeChat: cstutorcs

3. Recall that a **run** in a string is a maximal non-empty substring in which all symbols are equal. For example, the string `0001111000` consists of three runs: a run of three `0`s, a run of four `1`s, and another run of three `0`s.

   (a) Let L be the set of all strings in $\{0, 1\}^*$ in which every run of `0`s has odd length and every run of `1`s has even length. For example, L contains ε and `00000` and `0001111000`, but L does not contain `1` or `0000` or `110111000`.

   Describe both a regular expression for L and a DFA that accepts L.

Assignment Project Exam Help

https://tutorcs.com

   (b) Let L' be the set of all non-empty strings in $\{0, 1\}^*$ in which the *number* of runs is equal to the *length* of the first run. For example, L' contains `0` and `1100` and `0000101`, but L' does not contain `0000` or `110111000` or ε.

   WeChat: cstutorcs

   Prove that L' is not a regular language.

4. Your cousin Elmo is visiting you for Christmas, and he's brought a different card game. Like your previous game with Elmo, this game is played with a row of n cards, each labeled with an integer (which could be positive, negative, or zero). Both players can see all n card values. Otherwise, the game is almost completely different.

On each turn, the current player must take the leftmost card. The player can either keep the card or give it to their opponent. If they keep the card, their turn ends and their opponent takes the next card; however, if they give the card to their opponent, the current player's turn continues with the next card. In short, the player that does *not* get the ith card decides who gets the $(i + 1)$th card. The game ends when all cards have been played. Each player adds up their card values, and whoever has the higher total wins.

For example, suppose the initial cards are $[3, -1, 4, 1, 5, 9]$, and Elmo plays first. Then the game might proceed as follows:

- Elmo keeps the 3, ending his turn.
- You give Elmo the -1.
- You keep the 4, ending your turn.
- Elmo gives you the 1.
- Elmo gives you the 5.
- Elmo keeps the 9, ending his turn. All cards are gone, so the game is over.
- Your score is $1 + 4 + 5 = 10$ and Elmo's score is $3 - 1 + 9 = 11$, so Elmo wins.

Describe an algorithm to compute the highest possible score you can earn from a given row of cards, assuming Elmo plays first and plays perfectly. Your input is the array $C[1..n]$ of card values. For example, if the input is $[3, -1, 4, 1, 5, 9]$, your algorithm should return the integer 10.

5. **Prove** that each of the following languages is undecidable:

   (a) $\big\{ \langle M \rangle \;\big|\; M \text{ accepts } \texttt{RICESTHEOREM} \big\}$

   (b) $\big\{ \langle M \rangle \;\big|\; M \text{ rejects } \texttt{RICESTHEOREM} \big\}$ *[Hint: Use part (a), **not** Rice's theorem]*

Assignment Project Exam Help

https://tutorcs.com

WeChat: cstutorcs

6. There are n galaxies connected by m intergalactic teleport-ways. Each teleport-way joins two galaxies and can be traversed in both directions. Also, each teleport-way uv has an associated cost of $c(uv)$ galactic credits, for some positive integer $c(uv)$. The same teleport-way can be used multiple times in either direction, but the same toll must be paid every time it is used.

Judy wants to travel from galaxy s to galaxy t, but teleportation is rather unpleasant, so she wants to minimize the number of times she has to teleport. However, she also wants the total cost to be a multiple of 10 galactic credits, because carrying small change is annoying.

Describe and analyze an algorithm to compute the minimum number of times Judy must teleport to travel from galaxy s to galaxy t so that the total cost of all teleports is an integer multiple of 10 galactic credits. Your input is a graph $G = (V, E)$ whose vertices are galaxies and whose edges are teleport-ways; every edge uv in G stores the corresponding cost $c(uv)$.

*[Hint: This is **not** the same Intergalactic Judy problem that you saw in lab.]*

Assignment Project Exam Help

https://tutorcs.com

WeChat: cstutorcs

(scratch paper)

Assignment Project Exam Help

https://tutorcs.com

WeChat: cstutorcs

(scratch paper)

Assignment Project Exam Help

https://tutorcs.com

WeChat: cstutorcs

You may assume the following problems are 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 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?

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 an undirected graph G, is there a path in G that visits every vertex exactly once?

HamiltonianCycle: Given an undirected graph G, is there a cycle in G that visits every vertex exactly once?

DirectedHamiltonianCycle: Given an directed graph G, is there a directed 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?

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?

Super Mario: Given an $n \times n$ level for Super Mario Brothers, can Mario reach the castle?

You may assume the following languages are undecidable:

$$\text{SelfReject} := \big\{ \langle M \rangle \mid M \text{ rejects } \langle M \rangle \big\}$$

$$\text{SelfAccept} := \big\{ \langle M \rangle \mid M \text{ accepts } \langle M \rangle \big\}$$

$$\text{SelfHalt} := \big\{ \langle M \rangle \mid M \text{ halts on } \langle M \rangle \big\}$$

$$\text{SelfDiverge} := \big\{ \langle M \rangle \mid M \text{ does not halt on } \langle M \rangle \big\}$$

$$\text{Reject} := \big\{ \langle M, w \rangle \mid M \text{ rejects } w \big\}$$

$$\text{Accept} := \big\{ \langle M, w \rangle \mid M \text{ accepts } w \big\}$$

$$\text{Halt} := \big\{ \langle M, w \rangle \mid M \text{ halts on } w \big\}$$

$$\text{Diverge} := \big\{ \langle M, w \rangle \mid M \text{ does not halt on } w \big\}$$

$$\text{NeverReject} := \big\{ \langle M \rangle \mid \text{Reject}(M) = \varnothing \big\}$$

$$\text{NeverAccept} := \big\{ \langle M \rangle \mid \text{Accept}(M) = \varnothing \big\}$$

$$\text{NeverHalt} := \big\{ \langle M \rangle \mid \text{Halt}(M) = \varnothing \big\}$$

$$\text{NeverDiverge} := \big\{ \langle M \rangle \mid \text{Diverge}(M) = \varnothing \big\}$$