UNIQNAME (print clearly): ____________________

EECS 376: Foundations of Computer Science   Fall 2023, University of Michigan, Ann Arbor

EECS 376 Final Exam Solutions, Fall 2023

Instructions: This exam is closed book, closed notebook. No electronic devices are allowed. You may use two 8.5 × 11-inch study sheets. Make sure you are taking the exam at the time slot and the classroom you were assigned by the staff.

Any deviation from these rules will constitute an honor code violation. In addition, the staff reserves the right **not** to grade an exam taken in a violation of this policy.

The exam consists of **5** multiple-choice questions, **3** short-answer questions, and **3** longer-answer questions. The multiple choice questions may have more than one correct answer; ***mark all correct answers***. For the short- and long-answer sections, please write your answers clearly in the spaces provided. If you run out of room or need to start over, you may use the blank page at the end, but you **MUST** make that clear in the space provided for the answer. The exam has 7 pages printed on both sides, including this page and the blank page at the end.

You must leave all pages stapled together in their original order.

Honor pledge:
I have neither given nor received aid on this exam, nor have I concealed any violations of the Honor Code.
I will not discuss the exam with anyone before exam grades are released.
I attest that I am taking the exam at the time slot and the classroom I was assigned by the staff.

Signature: ____________________

PRINT YOUR NAME/UNIQNAME AS CLEARLY AS YOU POSSIBLY CAN:

Full Name: ____________________

Uniqname: ____________________

Questionnaire (NOT GRADED)

Answers to these questions will have no impact on your exam grade or final grade.

1. The percentage of lectures that I attended is roughly:

   □ 0-20%.
   □ 20-40%.
   □ 40-60%.
   □ 60-80%.
   □ 80-100%.

2. What class resources did you routinely use? (Mark all that apply.)

   □ Live Lectures
   □ Recorded Lectures
   □ Discussion Sections
   □ Piazza
   □ Office Hours
   □ Course Notes on eecs376.org

UNIQNAME (print clearly): ______________________________

Multiple Choice — 6 Points Each

In all multiple choice questions, fill in all correct boxes and no incorrect boxes.

1. Which of the following languages are *guaranteed* to be in NP?

   □ $L_1 \cap L_2$, where $L_1 \in \mathsf{NP}$ and L_2 is NP-complete.

   □ L, where it is known that $L \notin P$.

   □ L, where $L \leq_T L_{\text{HALT}}$, where $L_{\text{HALT}} = \{\langle M, x\rangle \mid \text{TM } M \text{ halts on input } x\}$.

   □ $L_1 \cup L_2$, where L_1 and L_2 are both in NP.

   □ L, where $\overline{L} \in \mathsf{NP}$.

2. Suppose language A is NP-Complete and language $B = \{x \in \{0,1\}^* \mid x \text{ is a palindrome}\}$. Which of the following are true?

   □ If $B \leq_p A$ then $P = NP$.

   □ If $A \leq_p B$ then $P = NP$.

   □ $A \leq_T B$.

   □ $B \leq_T A$.

3. Suppose an $O(m^2)$-time algorithm is discovered that, given a boolean 3CNF formula ϕ with n variables and m clauses, returns a satisfying assignment, if there is one. What are the *guaranteed* consequences of this? (MAX-CLIQUE is the problem: given $G = (V, E)$ to find the largest $U \subseteq V$ such that U forms a clique in G.)

   □ There is an $O(N^2)$-time algorithm for *every* problem in NP, where N is the length of the input.

   □ $L = \{x \in \{0,1\}^* \mid \; x = 0^n1^n \text{ for some } n \geq 0\}$ is NP-complete.

   □ $\mathsf{P} = \mathsf{NP}$.

   □ There is an efficient 99/100-approximation algorithm for MAX-CLIQUE.

   □ The Diffie-Hellman protocol can be broken in polynomial time.

4. Suppose A is a 1/2-approximation algorithm for the MAX-CLIQUE problem. Then

   □ $A(G)$ is guaranteed to return a clique in G of size $n/2$ if such a clique exists, where $G = (V, E)$ and $|V| = n$.

   □ $A(G)$ is guaranteed to return a clique of at least twice the maximum clique in G.

   □ $A(G)$ is guaranteed to return a clique of size at least half the largest clique in G.

   □ $A(G)$ *always* returns a larger clique than what a 1/4-approximation algorithm for MAX-CLIQUE would return.

5. Which of these decision problems is in the set NP?

   □ Given a graph G, decide if it has a vertex cover of size k.

   □ Given a set of distinct integers, decide if they can be partitioned into three sets with the same sum.

   □ Given a prime p, a generator g, and $t \in \{1, \ldots, p-1\}$, decide if the integer $i \in \{1, \ldots, p-1\}$ with $g^i \equiv t \pmod{p}$ is even.

   □ Given two integers p, q, decide whether their greatest common divisor is less than 10.

   □ Given a Turing machine M, decide if it halts when the input is the empty string.

UNIQNAME (print clearly): ____________________

Short Answer — 11 Points Each

1. Alice and Bob agree on the prime $p = 17$ and generator $g = 6$, and wish to establish a shared secret key k using the Diffie-Hellman protocol. Suppose Alice privately chooses $a = 3$ and Bob privately chooses $b = 12$. What is Alice's message to Bob? What is Bob's message to Alice? What is the secret key k?

2. Alice uses modulus $n = 65$ and RSA public key $e = 29$. Determine a possible private key d and calculate Alice's RSA-signature for the message $m = 3$.

3. You've invented a new, ultrafast randomized primality testing algorithm but it makes lots of mistakes, both false positives and false negatives. If n is prime, it reports "PRIME" with probability 3/4 and "COMPOSITE" with probability 1/4. If n is composite, it reports "COMPOSITE" with probability 3/4 and "PRIME" with probability 1/4. Explain how to reduce the error probability of this algorithm from 1/4 to any desired $\delta > 0$. How many times do you need to call the primality tester, as a function of δ? (Hint: consider returning a majority vote of the answers.)

UNIQNAME (print clearly): ____________________

Long Answer — 12 Points Each

1. Recall that a *tree* is a connected, acyclic graph, and that a *leaf* in a tree is a vertex with degree 1 (incident to 1 edge). The *minimum-leaf* problem is to find a spanning tree with the fewest number of leaves. We express this as a decision problem MIN-LEAF.

   $$\text{MIN-LEAF} = \{\langle G, k\rangle \mid \text{ undirected graph } G \text{ contains a spanning tree with at most } k \text{ leaves}\}$$

   Prove that MIN-LEAF is NP-hard by reducing HAM-PATH to MIN-LEAF.

   $$\text{HAM-PATH} = \Big\{\langle G', s, t\rangle \mid \text{ undirected graph } G' = (V', E') \text{ contains a Hamiltonian path from } s \text{ to } t,\ s, t \in V'.\Big\}$$

UNIQNAME (print clearly): ____________________

2. In the EQUITABLE-SAT problem we are given a list of clauses $\phi = (C_1, C_2, \ldots, C_m)$, where each clause C_j is a list of exactly 4 literals involving 4 distinct variables, say $(x, \overline{y}, \overline{w}, z)$. Given an assignment, we say that a clause is *equi-satisfied* if it contains *equal* numbers of TRUE and FALSE literals. For example, if x, y, w are TRUE and z is FALSE, $(x, \overline{y}, \overline{w}, z)$ would **not** be equi-satisfied because it contains one TRUE literal and three FALSE ones.

   Give a randomized approximation algorithm for finding an assignment to the variables that, in expectation, equi-satisfies a constant fraction $\rho \in [0, 1]$ of the clauses. Analyze what ρ is exactly. Give a lower bound on the probability that your algorithm satisfies at least a $\rho/2$-fraction of the clauses, using Markov's inequality.

UNIQNAME (print clearly): ____________________

3. SecureCo sells software to produce RSA keys. Every time you run SecureCo's `KeyGen()` procedure it randomly generates a tuple (n, e, d), where $n = p \cdot q$, $de \equiv 1 \pmod{\phi(n)}$, and p and q are 512-bit primes. Unfortunately there is a subtle flaw in the code of `KeyGen()`: p is a *fixed* 512-bit prime number, and q is a *random* 512-bit prime number. (Although p is fixed, you do not know what it is.)

   Suppose Alice uses `KeyGen()` to produce (n, e, d) and publicly announces (n, e), while Bob uses `KeyGen()` to produce (n', e', d') and publicly announces (n', e'). Explain how to *efficiently* decode *any* encrypted message that you intercept, which was encrypted with (n, e). You may assume $n \neq n'$. Recall that if $n = pq$, $\phi(n) = (p - 1)(q - 1)$.

This page intentionally left blank.

Be sure to write “answer continues on page 8” under your answer if you use this page.