UNIQNAME (print): _____

EECS 376 Final Exam, Winter 2024

Instructions:

This exam is closed book, closed notebook. No electronic devices are allowed. You may use two 8.5×11-inch study sheets (both sides) that you prepared yourself. The last few pages of the exam are scratch paper; you may not use your own. Make sure you are taking the exam at the time slot and location you were assigned by the staff. ***Please print your UNIQNAME at the top of each page.***

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

The exam consists of **12 multiple-choice** questions, **3 short-answer** questions, and **2 longer-answer** questions. For the short- and longer-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 pages at the end, but you **MUST** make that clear in the space provided for the answer. The exam has 14 pages printed on both sides, including this page and the blank pages at the end.

Leave all pages stapled together in their original order.

Sign the honor pledge below.

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 until noon on Thursday May 2nd (once every student in the class has taken the exam, including alternate times).

I attest that I am taking the exam at the time slot and the location I was assigned by the staff.

Signature: _____

Print clearly:

Full Name: _____

Uniqname: _____

This page is intentionally left blank.

UNIQNAME (print): _____

Multiple Choice: Select the one correct option

For each of the problems in this section, select **the one** correct option. Each one is worth 3 points; no partial credit is given.

1. Select the **correct description** for the following statement: assuming that $\mathsf{P} \neq \mathsf{NP}$, the language

   $$\textsc{VertexCoverBig} = \{G = (V, E) : G \text{ is an undirected graph with a vertex cover of size } |V| - 1\}$$

   is NP-complete.

   ◯ The statement is true.

   ◯ The statement is false.

   ◯ It is unknown whether the statement is true or false.

2. Suppose we independently flip 10000 *biased* coins, each resulting in a head with probability 1%. You want to bound the probability that at least 150 heads are flipped. Let X be the random variable representing the number of heads.

   Select the **tightest correct bound that can be obtained from either Markov's inequality or the Chernoff bound**, whichever applies.

   ◯ $\Pr[X \geq 150] \leq \frac{100}{150}$

   ◯ $\Pr[X \geq 150] \geq \frac{100}{150}$

   ◯ $\Pr[X \geq 150] \leq e^{-25/3}$

   ◯ $\Pr[X \geq 150] \leq e^{-50/3}$

   ◯ Neither Markov nor Chernoff applies to this setting.

3. Alice wants to use the RSA system with modulus $n = 33 = 3 \cdot 11$. Select the **private exponent d and public exponent e that are a valid choice for this modulus**.

   ◯ $d = 3, e = 22$

   ◯ $d = 9, e = 10$

   ◯ $d = 17, e = 2$

   ◯ $d = 3, e = 7$

   ◯ $d = 2, e = 17$

4. Select the problem that is ***not known*** **to admit an efficient deterministic algorithm**.

   ◯ Given an integer n, check if n is odd.

   ◯ Given positive integers m, i, n, compute $m^i \bmod n$.

   ◯ Given integers m, n, compute $m \cdot n$.

   ◯ Given $n = pq$ for some distinct primes p, q and $\phi(n) = (p-1)(q-1)$, compute the set $\{p, q\}$.

   ◯ Given positive integers m, t, n, compute an integer i such that $m^i \equiv t \pmod{n}$, if such an i exists.

5. Select the value of 3^{3333} mod 11.

   ◯ 1
   ◯ 3
   ◯ 5
   ◯ 6
   ◯ 9

6. Let $G = (V, E)$ be a *directed* graph with no self-loop edge. For any subset $S \subseteq V$ of vertices, let

$$E(S) = \{(u, v) \in E : u \in S, v \notin S\}$$

   denote the set of edges going from S to outside S.

   Suppose that, for each vertex $v \in V$, we independently include v in S with probability 1/2.

   Select the **expected value of** $|E(S)|$.

   ◯ $|E|/2$
   ◯ $|E|/4$
   ◯ $|V|/2$
   ◯ $|V|/4$
   ◯ 0

UNIQNAME (print): _____

Multiple Choice: Select <u>all</u> valid options

For each of the problems in this section, select <u>all</u> valid options; this could be all of them, none of them, or something in between.

The scoring for each problem is as follows: 5 points for all five correct (non-)selections; and 3, 2, 1, 0 points for four, three, two, one (or zero) correct (non-)selections, respectively.

1. Select all of the following that are **known *not* to exist**.
   - ☐ A language that is efficiently decidable and not efficiently verifiable.
   - ☐ A language that is undecidable and NP-Complete.
   - ☐ A language that is undecidable and NP-Hard.
   - ☐ A language that is not in P but is in NP.
   - ☐ A language that is in P and is NP-complete.

2. The *long*-path (decision) problem is: given an undirected unweighted graph G, vertices s and t, and a budget k, determine whether there exists a simple path in G from s to t having length *at least* k. (Recall that a simple path visits each vertex at most once.)

   The *short*-path (decision) problem is defined identically, but with "at most" in place of "at least."

   Select **all of the statements that are known to be true**.
   - ☐ The *long*-path problem is in NP.
   - ☐ The *long*-path problem is NP-hard, because there is a polynomial-time mapping reduction from the Hamiltonian-path problem to it.
   - ☐ There is a polynomial-time mapping reduction from the *short*-path problem to the *long*-path problem.
   - ☐ If P = NP, then the *short*-path problem is NP-complete.
   - ☐ If P ≠ NP, then the *short*-path problem is not NP-complete.

3. Suppose that A is a 2/3-approximation algorithm for the MaxClique problem, and let $G = (V, E)$ be the (undirected, unweighted) input graph. Select **all of the true statements**.
   - ☐ $A(G)$ must output a clique in G of size at least $2|V|/3$.
   - ☐ $A(G)$ must output a clique that has at least half as many vertices as a largest clique in G.
   - ☐ $A(G)$ must output a clique that has at least two-thirds as many vertices as a largest clique in G.
   - ☐ $A(G)$ cannot output a largest clique in G.
   - ☐ $A(G)$ must output a larger clique than $A'(G)$ does, where A' is a 1/2-approximation algorithm for MaxClique.

UNIQNAME (print): _____

4. Select **all of the statements that are known to be true**.

- ☐ Suppose that processes A and B, which might depend on each other, fail with probabilities p_a and p_b, respectively. Then $\Pr[\text{both processes succeed}] \geq 1 - p_a - p_b$.
- ☐ Let $X = \sum_{i=1}^{n} X_i$ for random variables $X_i \in [0, 1]$. Then $\Pr[X \geq 2\,\mathbb{E}[X]] \geq 1/2$.
- ☐ Let $X = \sum_{i=1}^{n} X_i$, where X_i is the indicator random variable for whether the ith U-Michigan student (out of a total of n) plays volleyball with their friends on the sand court outside Beyster, the day after finals are over. The Chernoff bound can be used to bound the probability that $X \geq 2\,\mathbb{E}[X]$.
- ☐ If you repeatedly toss a fair coin until you get a head, the expected number of coin tosses is infinite.
- ☐ If a program takes T steps in expectation, then it takes less than $4T$ steps with probability at least $3/4$.

5. Select **all of the statements that are known to be true.**

- ☐ If $\mathsf{P} = \mathsf{NP}$, then every NP-hard language is in P.
- ☐ If 3SAT is efficiently verifiable, then every language in NP is efficiently decidable.
- ☐ There *exists* an NP-hard language in P if and only if *every* NP-complete language is in P.
- ☐ HamiltonianCycle $\in \mathsf{P}$ if and only if SubsetSum $\in \mathsf{P}$.
- ☐ Clique $\notin \mathsf{P}$.

6. Select **all of the statements that are known to be true**.

- ☐ The one-time pad is information-theoretically secure even if the secret key is drawn from an arbitrary non-uniform distribution, as long as it is used only once.
- ☐ If there is an efficient algorithm for solving the discrete log problem, then there is an efficient algorithm for computing the shared secret key from the public messages in the Diffie-Hellman protocol.
- ☐ If there is an efficient algorithm for computing the shared secret key from the public messages in the Diffie-Hellman protocol, then there is an efficient algorithm for solving the discrete log problem.
- ☐ For all integers m, we have $(m^{27})^3 \equiv m \pmod{55}$.
- ☐ If a secure commitment scheme exists, then there is a zero-knowledge proof for 3SAT.

UNIQNAME (print): _____

Short Answers — 7 Points Each

1. Alice and Bob agree on the prime $p = 19$ and generator $g = 10$ of $\mathbb{Z}_p^*$, and wish to establish a shared secret key using the Diffie–Hellman protocol.

   Based on their secret exponents, Alice sends $x = 5$ and Bob sends $y = 17$.

   Derive one of their secret exponents and their shared secret key, filling in the blanks in the sentence below. All values should be from $\mathbb{Z}_{19}^* = \{1, \ldots, 18\}$.

   The secret exponent of _____ (**write one name**) is _____, and the shared secret key is _____.

   Briefly justify (in 1–2 sentences) your answers.

2. Recall that a *triangle* in an undirected graph $G = (V, E)$ is a set $T = \{x, y, z\} \subseteq V$ of three (distinct) vertices where $(x, y), (y, z), (x, z) \in E$.

   A *triangle cover* in G is a subset of vertices $C \subseteq V$ such that *every* triangle T in G has at least one vertex in C, i.e., $T \cap C \neq \emptyset$.

   The *minimum triangle cover* problem MinTriCover is: given G, find a triangle cover of minimum size, i.e., having as few vertices as possible. It is known that the decision version of this problem is NP-complete.

   Consider the following (greedy, polynomial-time) algorithm for approximating MinTriCover.

```
1: function GreedyTriCover(G)
2:     C ← ∅
3:     while there is some 'uncovered' triangle T in G where T ∩ C = ∅ do
4:         C ← C ∪ T, i.e., add all the vertices of T to C
5:     return C
```

   (a) **Correctly fill in the blank** in the following claim (you will justify your answer in the next parts). The ***smallest* approximation factor** $\alpha \geq 1$ that GreedyTriCover is *guaranteed* to obtain is

   $$\alpha = \_\_\_\_\_ .$$

   (b) **Draw a small input graph** for which the approximation factor α you gave in the previous part is *tight*, i.e., GreedyTriCover **necessarily obtains that factor, and no better**. Also, **briefly justify** (in 1-2 sentences) why this is so.

   (c) **Briefly prove** (in 3-4 sentences) that GreedyTriCover obtains the approximation factor α you gave in the first part, on any input graph.

UNIQNAME (print): _____

3. An instance of the MaxMod3 problem is m equations involving n variables $x_1, \ldots, x_n \in \mathbb{Z}_3 = \{0, 1, 2\}$. The jth equation has the form

$$a_j + b_j \equiv c_j \pmod 3 ,$$

where a_j and b_j are ***distinct variables*** from $x_1, \ldots, x_n$, and $c_j \in \mathbb{Z}_3$ is a ***constant***. The goal is to find an assignment to the variables that maximizes the number of satisfied equations.

For example, if $x_1 = 1$, $x_2 = 2$, $x_3 = 0$, then the equation $x_1 + x_2 \equiv 0 \pmod 3$ is satisfied, but $x_1 + x_3 \equiv 2 \pmod 3$ is not satisfied.

Consider the algorithm that assigns each variable x_i a uniformly random and independent value in $\mathbb{Z}_3$.

(a) **Correctly fill in the blank**: the expected number of satisfied equations is _____.

(b) **Rigorously justify** the answer you gave in the previous part using indicator random variables.

Long Answers

Question 1 is worth 14 points. Question 2 is worth 17 points.

1. Prove that the language

$$L_{\text{ExactFour}} = \{\langle M \rangle : M \text{ is a TM that accepts exactly four distinct inputs}\}$$

is undecidable, by reduction from either L_{ACC} or $L_{\varepsilon\text{-HALT}}$. Recall that their definitions are

$$L_{\text{ACC}} = \{(\langle M \rangle, x) : M \text{ is a Turing machine that accepts } x\} ,$$
$$L_{\varepsilon\text{-HALT}} = \{\langle M \rangle : M \text{ is a Turing machine that halts on } \varepsilon\} .$$

2. Consider the following one-player (solitaire) game, played on a finite rectangular grid of cells—which may have any number of rows and columns—and with stones that are each colored *black* or *white*.

   Initially, each cell has zero or more stones, which may be of the same or different colors. The goal of the game is to remove zero or more of the stones from the grid, so that:

   1. each row has only one color of stone, or no stone at all, and
   2. each column has at least one stone.

   Some initial setups can be solved (and may even have multiple solutions), while others cannot be solved.

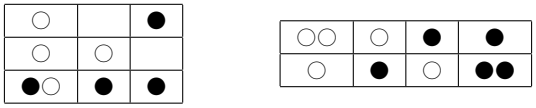

   Figure 1: In the above examples, ○ represents a white stone and ● represents a black stone. The first example is solvable by removing the bottom-left white stone and the top-right black stone. The second example cannot be solved.

   The Solitaire decision problem is: given an n-by-m grid with an initial setup of stones, determine whether it can be solved.

   (a) **Briefly justify** (in 2–3 sentences, without pseudocode) why Solitaire is in NP.

   (b) For showing that Solitaire is NP-hard, **define a polynomial-time mapping reduction** f from 3SAT to Solitaire, and **briefly justify its polynomial running time** (in 1–2 sentences). Do not address correctness; you will prove that in the next parts.

   Hint: Construct an initial setup where the rows correspond to the variables, the columns correspond to the clauses, and there are exactly three stones per column.

(c) **Prove** that $\phi \in$ 3SAT $\implies f(\phi) \in$ Solitaire for your reduction f.

(d) **Prove** that $f(\phi) \in$ Solitaire $\implies \phi \in$ 3SAT for your reduction f.

This page is intentionally left blank for scratch work.

If you have answers on this page, write "answer continues on page 13" in the corresponding solution box.

This page is intentionally left blank for scratch work.

If you have answers on this page, write "answer continues on page 14" in the corresponding solution box.