• You will have 75 minutes (1.25 hours) to solve 7 problems. Most have multiple parts. Don't spend too much time on questions you don't understand and focus on answering as much as you can!

• No resources are allowed for use during the exam except a multi-page cheatsheet and scratch paper on the back of the exam. *Do not tear out the cheatsheet or the scratch paper!* It messes with the auto-scanner.

• You should write your answers *completely* in the space given for the question. We will not grade parts of any answer written outside of the designated space.

• Please *use a dark-colored pen* unless you are absolutely sure your pencil writing is forceful enough to be legible when scanned. We reserve the right to take off points if we have difficulty reading the uploaded document.

• Unless otherwise stated, assume \( P \neq NP \).

• Assume that whenever the word "reduction" is used, we mean a (not necessarily polynomial-time) *mapping/many-one* reduction.

• You can only refer to the cheat sheet content as a black box.

• *Don't cheat.* If we catch you, you will get an F in the course.

• *Good luck!*

Name: _______________________________

NetID: ______________________________

Date: ______________________________
1 Short Answer I (8 questions) - 16 points

For each of the problems circle *true* if the statement is *always* true, circle *false* otherwise. There is no partial credit for these questions.

(a) If $A$ is a NP-Complete language and $B$ is a NP-Hard language, then $A \leq_{P} B$.

True    False

(b) If $A$ is a NP-Complete language then $A \leq_{P} SAT$ and $SAT \leq_{P} A$.

True    False

(c) If $A \leq_{P} B$ and $B$ is NP-Complete, then $A$ is NP-Complete.

True    False

(d) If $A \leq_{P} B$ and $A$ is NP-Complete, then $B$ is NP-Complete.

True    False

(e) Every decidable language is in NP.

True    False

(f) All NP problems are recursively enumerable.

True    False

(g) Every regular language is in P

True    False

(h) If $A$ and $B$ are both in NP, then $A \leq_{P} B$

True    False
2 Short Answer II (3 questions) - 9 points

For each of the problems circle all the answers that apply. There is no partial credit for these questions. Points are not necessarily divided evenly among all possible choices.

(a) Assume $X$ is the problem that finds the *hamiltonian cycle of minimum length* given a directed, weighted graph $G$. Circle the complexity classes this problem belongs to:

- $P$
- $NP$
- $NP$-hard
- $NP$-complete
- decidable
- undecidable

(b) The Tautology problem is the problem of determining if a 3SAT evaluates to true under every possible assignment to its variables. Tautology belongs to:

- $P$
- $NP$
- $NP$-hard
- $NP$-complete
- decidable
- undecidable

(c) Recall the primality problem is problem of determining if a number ($n$) is prime (has factors $< n$). Primality belongs to:

- $P$
- $NP$
- $NP$-hard
- $NP$-complete
- decidable
- undecidable
3 Classification I (P/NP) - 15 points

Is the following problem in P, NP, or some combinations of complexity classes? For each of the following problems, circle all the complexity classes that problem belongs to. Whatever class it is in, prove it!

The 374 path problem (374P) asks given an undirected graph $G$, does $G$ contain a path that visits at least 374 vertices.

- **INPUT:** A graph $G$.
- **OUTPUT:** TRUE if there exists a path that is at least 374 vertices long. FALSE otherwise.

Which of the following complexity classes does this problem belong to? Circle all that apply:

- P
- NP
- NP-hard
- NP-complete
4 Classification II (P/NP) - 15 points

Is the following problem in P, NP, or some combinations of complexity classes? For each of the following problems, circle all the complexity classes that problem belongs to. Whatever class it is in, prove it!

The multi-solution SAT (MultiSAT) problem asks whether a SAT problem has multiple satisfiable truth assignments.

- **INPUT:** A SAT formula \( \phi \).
- **OUTPUT:** True if there exists at least two distinct variable assignments that satisfy this formula. False otherwise.

Which of the following complexity classes does this problem belong to? Circle all that apply:

\[
P \quad NP \quad NP-hard \quad NP-complete
\]
5 Classification III (P/NP) - 15 points

Is the following problem in P, NP, or some combinations of complexity classes? For each of the following problems, circle all the complexity classes that problem belongs to. Whatever class it is in, prove it!

HALT$^T_M$ you are given a turing machine $\langle M \rangle$ and must determine if it halts on a empty input.

- **Input:** A TM $\langle M \rangle$.
- **Output:** True if the will halt on an empty input. False otherwise.

Which of the following complexity classes does this problem belong to? Circle all that apply:

$$P \quad NP \quad NP\text{-hard} \quad NP\text{-complete}$$

You must justify (prove) your answer!
6 Classification I (Decidability) - 15 points

Are the following languages decidable? For each of the following languages,

- Circle one of "decidable" or "undecidable" to indicate your choice.

- If you choose "decidable", prove your choice correct by describing an algorithm that decides that language. If you choose "undecidable", prove your choice correct by giving a reduction proving its correctness.

- Regardless of your choice, explain briefly (i.e., in 3 sentences maximum, diagrams, clear pseudo-code) why the proof of the choice you gave is valid.

\[
\text{REACH}_{TM} = \{ (M, w, q) \mid M \text{ is a } TM = (Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej}) \text{ and will enter state } q \in Q \text{ on input } w \}
\]

\[
\Sigma = \{0, 1\}
\]

decidable undecidable
7 Classification II (Decidability) - 15 points

Are the following languages decidable? For each of the following languages,

- Circle one of "decidable" or "undecidable" to indicate your choice.

- If you choose "decidable", prove your choice correct by describing an algorithm that decides that language. If you choose "undecidable", prove your choice correct by giving a reduction proving its correctness.

- Regardless of your choice, explain briefly (i.e., in 3 sentences maximum, diagrams, clear pseudo-code) why the proof of the choice you gave is valid.

\[ \text{ACCEPT}_{374}^{TM} = \{ (M, w) \mid M \text{ is a } TM \text{ and accepts } w \text{ in 374 steps. } \} \]
\[ \Sigma = \{0,1\} \]

decidable  undecidable
This page is for additional scratch work!
Turing Machines

Turing machine is the simplest model of computation.
- Input written on (infinite) one sided tape.
- Special blank characters.
- Finite state control (similar to DFA).
- Every step: Read character under head, write character out, move the head right or left (or stay).
- Every TM \( M \) can be encoded as a string \( \langle M \rangle \).

Transition Function: \( \delta : Q \times \Gamma \to Q \times \Gamma \times \{\leftarrow, \rightarrow, \square\} \)

\( \delta(q,c) = (p,d,\leftarrow) \)
- \( q \): current state.
- \( c \): character under tape head.
- \( p \): new state.
- \( d \): character to write under tape head.
- \( \leftarrow \): Move tape head left.

Complexity Classes

<table>
<thead>
<tr>
<th>Computational Complexity Classes</th>
<th>Turing-unrecognizable</th>
<th>Turing-recognizable</th>
<th>Semi-Decidable</th>
<th>Decidable (Recursive)</th>
<th>Context-Free</th>
<th>Context-Sensitive</th>
<th>Context-Sensitive</th>
<th>Non-context-Free</th>
<th>Undecidable</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Algorithmic Complexity Classes (assuming P ( \neq ) NP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( NP )</td>
</tr>
<tr>
<td>( NPC )</td>
</tr>
<tr>
<td>( co-NP )</td>
</tr>
</tbody>
</table>

Reductions

A general methodology to prove impossibility results.
- Start with some known hard problem \( X \).
- Reduce \( X \) to your favorite problem \( Y \).

If \( Y \) can be solved then so can \( X \) \( \implies \) \( Y \). But we know \( X \) is hard so \( Y \) has to be hard too. On the other hand if we know \( Y \) is easy, then \( X \) has to be easy too.

The Karp reduction, \( X \leq \text{P} \text{.} Y \) suggests that there is a polynomial time reduction from \( X \) to \( Y \).

Sample NP-complete problems

- **3SAT**: Given a boolean formula in conjunctive normal form, with exactly three distinct literals per clause, does the formula have a satisfying assignment?
- **INDEPENDENTSET**: Given an undirected graph \( G \) and integer \( k \), what is there a subset of vertices \( \geq k \) in \( G \) that have no edges among them?
- **CLIQUE**: Given an undirected graph \( G \) and integer \( k \), is there a complete subgraph of \( G \) with more than \( k \) vertices?
- **3COLOR**: Given an undirected graph \( G \), can its vertices be colored with three colors, so that every edge touches vertices with different colors?
- **HAMPATH**: Given graph \( G \) (either directed or undirected), is there a path in \( G \) that visits every vertex exactly once?
- **LONGESTPATH**: Given a graph \( G \) (either directed or undirected, possibly with weighted edges) and an integer \( k \), does \( G \) have a path \( \geq k \) length?

Sample undecidable problems

- **ACCEPTONINPUT**: \( A_{TM} = \{ \langle M, w \rangle \mid M \text{ is a TM and } M \text{ accepts on } w \} \)
- **HALTONINPUT**: \( Halt_{TM} = \{ \langle M, w \rangle \mid M \text{ is a TM and } M \text{ halts on input } w \} \)
- **HALTONBLANK**: \( Halt_{TM} = \{ \langle M \rangle \mid M \text{ is a TM and } M \text{ halts on blank input} \} \)
- **EMPTINESS**: \( E_{TM} = \{ \langle M \rangle \mid M \text{ is a TM and } L(M) = \emptyset \} \)
- **EQUALITY**: \( EQ_{TM} = \{ \langle M_A, M_B \rangle \mid M_A \text{ and } M_B \text{ are TMs and } L(M_A) = L(M_B) \} \)