Problem 1. Is the following problem decidable? Give a proof.

Problem 100-TRANSITIONS1:

Instance: A TM $M$;
Query: Is there an input $w$ on which $M$ makes more than 100 transitions?

Problem 2. Which of the following problems is decidable? Give proofs.

PROBLEM A:

Instance: A Turing machine $M$ and a word $w$;
Query: Is there a state of $M$ which is entered at least twice when running $M$ on $w$?

PROBLEM B:

Instance: A Turing machine $M$ and a word $w$;
Query: Is there a state of $M$ which is entered twice in a row when running $M$ on $w$?

Problem 3. Is the following problem decidable? Give a complete proof.

Problem 3WORDS:

Instance: A Turing machine $M$;
Query: Is $L(M) = \{\epsilon, 0, 00\}$?

Problem 4. Prove that a language $L$ is decidable if and only if $L$ can be enumerated in lexicographic order.

Problem 5. Is the following problem decidable? Give a proof.

Problem ASTAR:

Instance: A TM $M$;
Query: Is $L(M) = a^*$?

Problem 6. Is the following problem decidable: “Give a context-free grammar $G$, does $G$ generate all words in $\Sigma^*$ of even length?”.

Problem 7. Is the following problem decidable? Give a proof.

SINGLETON:

Instance: A TM $M$;
Query: Is $|L(M)| = 1$, that is, does $M$ accept exactly one word?

**Problem 8.** Is the following problem decidable? Give a proof.

*Instance:* A Turing machine $M$;

*Query:* Does there exist a word $w$ such that in the computation on $w$ machine $M$ halts with the blank tape?

**Problem 9.** Is the following problem decidable:

*Instance:* A TM $M$ and a word $w$

*Query:* Does $M$ ever moves to the left in the computation on $w$?

**Problem 10.** Is the following problem decidable:

*Instance:* A TM $M$

*Query:* Is there a word $w$, such that in the computation on $w$ $M$ moves to the left at least once?

**Problem 11.** Is the following problem decidable? Give a complete proof.

*Repeated State:*

*Instance:* A Turing Machine $M$, an input $w$;

*Query:* Does $M$ use all its states in the computation on $w$?

**Problem 12.** Is the following problem decidable? Give a proof.

*Instance:* A Turing machine $M$;

*Query:* Does $L(M)$ contain a word of even length?

**Problem 13.** Let $M$ be a Turing machine which never moves to the left. Prove that $L(M)$ is regular.

**Problem 14.** Briefly define or state.

1. The Halting Problem
2. Savitch’s Theorem
3. The Satisfiability Problem and Cook’s Theorem
4. The Church Thesis

**Problem 15.** For each of the problems below, tell if the problem is

- decidable,
• undecidable but recursively enumerable,
• not recursively enumerable.

Write "r" if the problem is decidable, "nr" if it's r.e. but not decidable, or "non r.e." if it's not recursively enumerable. M denotes a Turing machine. No proofs are necessary.

(a) \( L(M) \) contains exactly five words.
(b) \( L(M) \) contains a word in \((ab)^*\).
(c) All words in \( L(M) \) end with an a.
(d) \( L(M) \) contains a word of length 3 or 7 or 19.

**Problem 16.** Given two strings \( u = a_1a_2\ldots a_n \) and \( v = b_1b_2\ldots b_m \), the shuffle of \( u, v \), denoted \( u \oplus v \) is the set of all strings that can be obtained by interspersing letters from \( u \) and \( v \) in a way that maintains the left-to-right order of those letters in each string. More precisely, if \( w = c_1\ldots c_{n+m} \), then \( w \in u \oplus v \) iff there are two disjoint sequences of indices \( i_1 < i_2 < \ldots < i_n, j_1 < j_2 < \ldots < j_m \), such that \( c_i = a_i \) and \( c_j = b_j \). For example:

\[
ab \oplus ca = \{abca, acba, acab, caba, caab\}
\]

Given two languages \( L_1, L_2 \), the shuffle of \( L_1 \) and \( L_2 \) is the set of all shuffles of pairs of words in \( L_1 \) and \( L_2 \), that is

\[
L_1 \oplus L_2 = \{u \oplus v : u \in L_1 \& v \in L_2\}
\]

(a) Prove that the class of recursive languages is closed under the shuffle operation.
(b) Prove that the class of r.e. languages is closed under the shuffle operation.

Please make sure that the descriptions of the TMs you construct are clear and unambiguous. You can use either plain English, or a Pascal-like notation.

**Problem 17.** Prove that the class of recursively enumerable languages is closed under the Kleene star operation. In other words, you have to show that if \( L \) is r.e. then so is \( L^* \).

**Problem 18.** Is the following decision problem decidable:

*Instance*: A TM \( M \), a word \( w \)

*Query*: Does \( M \) ever makes three consecutive moves to the right in the computation on input \( w \)?

Please make the proof clear and precise.

**Problem 19.** Is the following problem decidable? Give a proof.

Problem BackToMarker:
**Problem 20.** Which of the two following problems is decidable. Give proofs.

**Problem LongWords:**

*Instance:* A context-free grammar $G$ with alphabet $\Sigma$;

*Query:* Does $G$ generate all words over $\Sigma$ of length $\geq 100$?

**Problem ShortWords:**

*Instance:* A context-free grammar $G$ with alphabet $\Sigma$;

*Query:* Does $G$ generate all words over $\Sigma$ of length $\leq 100$?

**Problem 21.** Are the following problems (a) Decidable? (b) Not decidable, but r.e.? (c) Not r.e.? Give proofs. To show decidability sketch the algorithm. For full credit do not use Rice’s theorems. Partial credit will be given if you do use Rice’s theorems.

**Problem P1:**

*Instance:* A TM $M$;

*Query:* Is $L(M) \subseteq a^*$?

**Problem P2:**

*Instance:* A cfg $G$.

*Query:* Is $L(G) \subseteq a^*$?

**Problem 22.** Are the following problem decidable?

**Problem CFLODD:**

*Instance:* A cfg $G$;

*Query:* Does $L(G)$ contain a word of odd length?

**Problem CFLALLODD:**

*Instance:* A cfg $G$;

*Query:* Are all words in $L(G)$ of odd length?

**Problem ALLODDCFL:**
Instance: A cfg G;
Query: Are all odd words over $\Sigma$ in $L(G)$?

Problem 23. Give a complete, direct proof that the following problem $\mathcal{P}$ is undecidable:

Instance: A Turing machine $M$.
Query: Does $M$ accept $abba$?

Problem 24. (a) Define Hilbert’s 10th Problem.
(b) A degree-2 polynomial is a polynomial of the form $p(x_1, \ldots, x_k) = \sum_{i \leq j} a_{ij} x_i x_j + \sum_i b_i x_i + c$. Prove that the following problem is undecidable, by showing that Hilbert’s 10th Problem reduces to it.

SQDE:

Instance: A set of degree-2 polynomials $p_i(x_1, \ldots, x_k)$, for $i = 1, \ldots, n$, with integer coefficients.
Query: Are there integers $e_1, \ldots, e_k$ such that $p_i(e_1, \ldots, e_k) = 0$ for each $i = 1, \ldots, n$?

Problem 25. For each of the two statements (i) and (ii) below tell whether it is true or false and justify your answer.

(i) There is a Turing machine $M$ for which the decision problem “Given $w$, does $M$ accept $w$?” is undecidable.
(ii) For each Turing machine $M$ the decision problem “Given $w$, does $M$ accept $w$?” is undecidable.

Problem 26. Prove that the class of r.e. languages is closed under concatenation and Kleene’s star, that is

(a) if $L_1, L_2$ are r.e. then $L_1L_2$ is r.e.
(b) if $L$ is r.e. then so is $L^*$.

Please make sure that the descriptions of the TMs you construct are clear and unambiguous. You can use either plain English, or a Pascal-like notation.

2 Post Correspondence Problem

The instances of the PCP below are in a different format than what we use in class this year. In some problems they are specified by two lists, in other by a list of pairs. The ”solution” of a PCP means a match.

Problem 1. Does the following instance of PCP have a solution? Give a proof.

$$ A = (ba, baab, ab, a, ab, aab) $$

$$ B = (a, ba, abaa, bb, aaba, b) $$

Problem 2. Which of the following instances of PCP have solutions. Give proofs.
Problem 3. Let UPCP be the restriction to PCP in which all strings are over a one letter alphabet. Prove that UPCP is decidable.

Problem 4. Let PCP1 be the restriction of PCP in which the alphabet is \( \{a, b, c\} \), and each string contains exactly one letter \( c \) (the numbers of \( a \)'s and \( b \)'s are arbitrary). Is PCP1 decidable?

Problem 5. Does the following instance of PCP have a solution? Give a proof.
\[
A = (aab, bba, baa, babb, a, aaba) \\
B = (baa, b, ba, bab, aaa, ab)
\]

Problem 6. Which of the following instances of PCP have solutions? Show the solution, otherwise prove it does not exist.

1. \( A = (aa, bb, abb), B = (aab, ba, b) \).
2. \( A = (ab, aa, ab, bb), B = (bb, ba, abb, bab) \).
3. \( A = (ab, baa, aba), B = (aba, aa, baa) \).
4. \( A = (a, aab, abab), B = (aaa, bab, ab) \).

Problem 7. Which of the following instances of the PCP have matches?
\[
P_1 = (aba, a, [bab], [b], [ab], [aba, bab]) \\
P_2 = (aba, a, [bab], [b], [abba], [bab, bab]) \\
P_3 = (aba, a, [bab], [b], [bb], [bab, bab]) \\
P_4 = (ba, ba, [a], [bab], [bab, bab]) \\
P_5 = (aba, a, [bab], [bab], [aba, bab])
\]

Problem 8. Let PCP1 be a version of PCP in which every string on input is a binary string with exactly one 1. For example, an instance of PCP1 might look like this:
\[
P = ([0100, 0101, 00010], [0001, 01])
\]
Is PCP1 decidable? Give a complete proof.

Problem 9. Which of the following instances of PCP have solutions:

\[
A = (10, 1001, 1, 110) \\
B = (11010, 10, 0110, 011)
\]

\[
A = (0101, 010, 1, 101, 1, 110) \\
B = (01, 1, 000, 10, 0, 1101)
\]
(a) \( A = (ab, bbaaba, b, bb) \), \( B = (a, a, bbbb, ab) \)

(b) \( A = (b, ba, a, baa) \), \( B = (ba, ab, abb, ab) \)

(c) \( A = (ab, bbaa, b, abba) \), \( B = (abb, a, ba, baa) \)

(d) \( A = (abaa, ba, b) \), \( B = (ab, a, ba) \)

Give a solution (if it exists), or prove that it does not exist.

**Problem 10.** Is the following problem decidable? Give a complete proof.

**PseudoPCP:**

- **Instance:** Two sets of strings \( x_1, \ldots, x_n, y_1, \ldots, y_n \);
- **Query:** Are there two sequences if indices \( i_1, \ldots, i_p, j_1, \ldots, j_q \) such that
  \[
  x_{i_1}x_{i_2} \cdots x_{i_p} = y_{j_1}y_{j_2} \cdots y_{j_q}.
  \]

3 **NP-Completeness and Related Topics**

**Problem 1.** If \( \Sigma \) is an alphabet, by \( \Sigma^n \) we denote the strings in \( \Sigma^* \) of length \( n \). If \( v \) is a string then by \( v[i] \) we denote the \( i \)-th symbol of \( v \). Prove that the following problem is NP-complete:

**STRING**

- **Instance:** A finite set of strings \( S \subseteq \{0, 1, a\}^n \).
- **Query:** Is there a string \( w \in \{0, 1\}^n \) such that for each \( v \in S \) there is \( i \in \{1, 2, \ldots, n\} \) for which \( w[i] = v[i] \)? In other words, \( w \) agrees with each string in \( S \) on at least one position.

**Problem 2.** A cycle \( H \) in a graph is called quasi-hamiltonian if, for each vertex \( v \), either \( v \) belongs to \( H \) or all its neighbors belong to \( H \). Prove that the problem below is NP-complete.

**QHC**

- **Instance:** A graph \( G = (V, E) \);
- **Query:** Does \( G \) have a quasi-hamiltonian cycle?

**Problem 3.** A feedback vertex set in a graph \( G \) is a set \( F \) vertices such that each cycle contains at least one vertex from \( F \). In other words, after removing \( F \) from \( G \) we obtain an acyclic graph. Prove that the following problem is NP-complete.

**FVS**

- **Instance:** graph \( G = (V, E) \), integer \( K \);
Query: Does $G$ contain a feedback vertex set of size at most $K$?

One of the following problems gives a simple reduction: 3CNF, VERTEX COVER, PARTITION.

**Problem 4.** Consider the following version of the satisfiability problem:

**2XOR-SAT**

*Instance:* A boolean expression $\alpha = X_1 \land X_2 \land \ldots \land X_n$, where each $X_i$ is of the form $u \oplus v$, for some literals $u, v$.

*Query:* Is $\alpha$ satisfiable?

The symbol $\oplus$ is the exclusive-or operation, that is $x \oplus y = 1$ iff $x \neq y$.

Is 2XOR-SAT

1. in P?
2. NP-complete?
3. Neither of the above?

Give a complete proof.

**Problem 5.** A bipartite graph is defined as $G = (X, Y, E)$, where $X \cup Y$ is a vertex set and $E$ is a set of edges $(x, y)$, where $x \in X$ and $y \in Y$. Prove that the following problem is NP-complete:

**BIPARTITE HAMILTONIAN CYCLE:**

*Instance:* A bipartite graph $G = (X, Y, E)$;

*Query:* Does $G$ have a hamiltonian cycle?

**Problem 6.** A set $D$ of vertices in a graph $G$ is called a dominating set if for each vertex $u$ either $u \in D$ or $u$ has a neighbor in $D$. Prove that the following problem is NP-complete.

**DOMINATING SET:**

*Instance:* A graph $G = (V, E)$, number $K$;

*Query:* Does $G$ have a dominating set of size $D$?

**Problem 7.** Consider the following decision problems:

**HAMILTONIAN PATH**

*Instance:* A graph $G = (V, E)$;

*Query:* Does $G$ have a hamiltonian path?

**HAMILTONIAN CYCLE**
**Instance:** A graph \( G = (V, E); \)

**Query:** Does \( G \) have a hamiltonian cycle?

Prove that HAMILTONIAN CYCLE reduces in log-space to HAMILTONIAN PATH. (You need to give an explicit reduction.)

Suggestion: The reduction in the opposite direction is a bit simpler, so you may try it first, for a warmup.

**Problem 8.** Prove that VERTEX COVER can be solved in polynomial time for trees (cycle-free graphs).

For full credit, your algorithm needs to run in linear time. Justify its correctness.

**Problem 9.** Prove that the following Dynamic Storage Allocation problem is NP-complete.

**DSA:**

**Instance:** A collection \( C \) of \( n \) triples \((s_i, a_i, d_i), i = 1, \ldots, n\), and an integer \( D \).

Each triple \((s_i, a_i, d_i)\) represents a memory request in a computer system: “from time \( a_i \) to time \( d_i \) I will need \( s_i \) memory locations”. We refer to numbers \( s_i, a_i, d_i \), respectively, as size, arrival time and departure time.

**Query:** Is there a feasible memory allocation for \( C \) into memory of size \( D \)?

A feasible allocation in size-\( D \) memory is a function \( f : \{1, \ldots, n\} \to \{1, \ldots, D\} \) that assigns to each request \( i \) address \( f(i) \) such that \( f(i) + s_i - 1 \leq D \), and such that no two jobs overlap. In other words, we require that if time intervals \([a_i, d_i], [a_j, d_j]\) overlap then memory blocks \([f(i), f(i) + s_i - 1]\) and \([f(j), f(j) + s_j - 1]\) don’t overlap.

**Problem 10.** Consider a version of the DSA problem in which all jobs have size 1 (that is \( s_i = 1 \) for \( i = 1, \ldots, n \)). Give an efficient algorithm that given such an instance of DSA computes an optimal memory allocation. The algorithm gets on input pairs \([a_i, d_i]\) and outputs memory allocation \( f \) that minimizes memory usage. The faster algorithm the better.

**Problem 11.** Prove that the following problem is NP-complete.

**Deg3ST:**

**Instance:** A graph \( G \);

**Query:** Does \( G \) have a spanning tree in which every vertex has degree \( \leq 3 \)?

Hint: A hamiltonian path is a spanning tree in which the degree of every vertex is \( \leq 2 \).

**Problem 12.** Prove that the following problem is NP-complete.

**MINIMUM COVER:**

**Instance:** A collection \( C \) of finite sets, an integer \( K \leq |C| \).

**Query:** Does \( C \) contain a cover of size at most \( K \), i.e. a subcollection \( C' \subseteq C \) such that \( |C'| \leq K \) and \( \bigcup_{X \in C'} X = S \)? (In other words, each \( x \in S \) belongs to at least one set in \( C' \).)
Hint: Use VERTEX COVER.

**Problem 13.** A “greedy” algorithm for an optimization problem considers only local properties of a given instance, and constructs a solution piece by piece by adding to it pieces that “look good” based only on local information. Below you are given greedy algorithms for three NP-complete problems. For each of those algorithms prove that, in general, it may not find an optimal solution.

**Some Definitions:** Let $G = (V, E)$ be a graph. $N(v)$ denotes the set of neighbors of $v$. If $X \subseteq V$ is a set of vertices, then $G[X]$ denotes the subgraph *induced* by $X$, that is a graph $G' = (X, E')$, where $E'$ contains all edges from $E$ between vertices in $X$.

**CLIQUE-OPT:** Given a graph $G$, find a largest clique in $G$.

Algorithm GreedyClique:

```
begin
    C := nil;
    repeat
        pick a vertex $v$ of $G$ of largest degree;
        C := C + {v};
        V := N(v);
        G := G[V];
    until V = empty;
    output(C)
end
```

**IND-SET-OPT:** Given a graph $G$, find a largest independent set in $G$.

Algorithm GreedyIndSet:

```
begin
    I := empty;
    repeat
        pick a vertex $v$ of $G$ of smallest degree;
        I := I + {v};
        V := V - {v} - N(v);
        G := G[V];
    until V = empty;
    output(I)
end
```

**VERTEX-COVER-OPT:** Given a graph $G$, find a smallest vertex cover of $G$.

Algorithm GreedyVCover:

```
begin
    U := empty;
    repeat
        pick a vertex $v$ of $G$ of largest degree;
```
U := U + {v};
V := V - {v};
G := G[V];
until E = empty;
output(U)

Problem 14. In the tiling problem we are given \( n \) rectangles called tiles, \( t_i = (w_i, h_i) \), \( i = 1, \ldots, n \), where \( w_i \) is the width and \( h_i \) is the height of \( t_i \). We are also given a floor rectangle \( F = (W, H) \), of width \( W \) and height \( H \). Our goal is to “tile” \( F \) with the given set of tiles, that is, to cover \( F \) with tiles, so that (a) no two tiles overlap, (b) each tile can be chosen only once, (c) each tile’s edges must be parallel to the edges of \( F \). We do not require that all tiles be used.

Prove that the following problem is NP-complete:

Problem TILING:

Instance: A set of tiles \( t_i = (w_i, h_i), i = 1, \ldots, n \), a floor rectangle \( F = (W, H) \);

Query: Can \( F \) be tiled with \( t_1, \ldots, t_n \)?

Problem 15. TILING2 is defined similarly to TILING (previous problem), except that we require that all times be used. Prove that TILING2 is NP-complete.

Problem 16. Let PRIMES be the set of binary representations of prime numbers. (a) Show that PRIMES \( \in \) coNP. (b) Show that PRIMES \( \in \) NP. (For part (b) you only need to present Pratt’s algorithm for PRIMES and state the number-theoretic result that implies its correctness. You do not need to analyze the time complexity.)

4 Classes P, L, NL and PSPACE

Problem 1. Give a polynomial space algorithm for the following problem.

FA INTERSECTION:

Instance: A collection of dfa’s \( M_1, M_2, \ldots, M_n \).

Query: Is \( L(M_1) \cap L(M_2) \cap \ldots L(M_n) = \emptyset \)?

Hint: Consider a “vector” machine \( M \), whose states are \( n \)-tuples \( (q^1, q^2, \ldots, q^n) \), where \( q_i \) is the state of \( M_i \). We have that \( L(M_1) \cap L(M_2) \cap \ldots L(M_n) \neq \emptyset \) iff there is a \( w \) that is accepted by all \( M_i \)’s. Call such \( w \) a witness. If \( m \) is the maximum number of states in each \( M_i \), then how long a shortest witness can be in terms of \( m \)?

Problem 2. Give an algorithm for REACHABILITY that works in space \( O(\log^2 n) \). State the algorithm in pseudo-code and justify the space bound. What is the time complexity of your algorithm?

Problem 3. Prove that ACYCLIC REACHABILITY is NL-complete, by proving that REACHABILITY \( \leq_L \) ACYCLIC REACHABILITY.
ACYCLIC REACHABILITY

Instance: An acyclic digraph \( D = (V, A) \), \( s, t \in V \);
Query: Is there a directed path from \( s \) to \( t \) in \( D \)?

Problem 4. Discuss the complexity of the following decision problem:

Problem NFA EMPTINESS:

Instance: A non-deterministic finite automaton \( M \);
Query: Is \( L(M) \neq \emptyset \)?

Find the smallest complexity class, among those discussed in class, to which it belongs. Is it complete for this class? Give proofs.

Problem 5. Prove that the following problem is P-complete:

CLF EMPTY

Instance: A context-free grammar \( G \);
Query: Is \( L(G) = \emptyset \)?

Problem 6. (a) Give a polynomial-space algorithm for QSAT (quantified boolean expressions). Describe the algorithm in pseudocode, as a recursive procedure.

(b) Which of the two following quantified boolean expressions is true?

\[
\forall x \exists y \forall z \ (x \land (y \lor z)) \lor (\bar{x} \land (\bar{z} \lor y))
\]

\[
\forall x \exists y \forall z \exists t \ (x \Rightarrow (y \land z)) \lor (\bar{x} \land (((z \Rightarrow t) \land y) \Rightarrow \bar{x}))
\]

Problem 7. Prove that CIRCUIT VALUE is P-complete for monotone circuits.

Problem 8. Let AGG be the restriction of GG (Generalized Geography) to acyclic directed graphs. Discuss the time and space complexity of AGG. Determine the smallest complexity class among L, NL, P, NP, PSPACE that (you think) AGG belongs to. Is AGG complete in this classes? Give complete proofs.

Problem 9. Let \( Q \) be the language of properly nested parentheses. For example, \( ()(()()) \in Q \) but \( ()(())) \notin Q \). Prove that \( Q \in L \).

5 True/False and Other Problems

Problem 1. True/False. Correct = 1, incorrect = -1, no answer = 0.
Y N All recursive languages are r.e.
Y N The complement of an r.e. language is r.e.
Y N The complement of a recursive language is r.e.
Y N The complement of a context-free language is context-free.
Y N If \( L \) and \( L' \) are r.e., then \( L \) is recursive.
Y N It is decidable, whether a pushdown automaton \( M \) accepts a given word \( w \).
Y N If \( L_1, L_2 \) are cfl's, then so is \( L_1 \cap L_2 \).
Y N There are cfl's \( L_1, L_2 \) such that \( L_1 \cap L_2 \) is a cfl.
Y N It is decidable, whether \( L(G) = \emptyset \), for a cfg \( G \).
Y N It is decidable, whether \( L(G) = \Sigma^* \), for a cfg \( G \).

Problem 2. Classify the languages below into one of the following categories:

- regular
- context-free (but not regular)
- recursive (but not context-free)
- r.e. (but not recursive)

No proofs are necessary. Negative credits may be given for serious errors.

1. \( L = \) the set of all encodings of Turing Machines with more than 10 states
2. \( L = \) the set of all encodings of Turing Machines that accept the empty language
3. \( L = \) the set of all encodings of Turing Machines that accept a non-empty language
4. \( L = \{ucv \mid u,v \in \{a,b\}^* \& u \text{ is a substring of } v\} \)
5. \( L = \{ucv^a \mid u,v \in \{a,b\}^* \& u \text{ is a substring of } v\} \)
6. \( L = \) the set of all binary encodings of positive integers that are divisible by 17

Problem 3. True/False.

Y N If \( L_1 \) is recursive and \( L_2 \) is context-free then \( L_1 \cap L_2 \) is recursive.
Y N If \( L_1 \) is recursive, and \( L_1 \subseteq L_2 \) then \( L_2 \) is r.e.
Y N If \( L_1 \) is recursive, and \( L_2 \) is regular then \( L_1 \cap L_2 \) is regular.
Y N If \( L_1 \) is regular and \( L_2 \) is context-free then \( L_1 \cap L_2 \) is regular.
Y N It is decidable, whether \( w \in L(M) \) for a pda \( M \).
Y N If \( L = L_1 \cap L_2 \), and \( L \) is not r.e., then one of \( L_1 \), \( L_2 \) is not r.e.
Y N If \( L = L_1 \cup L_2 \) and \( L \) is recursive, then one of \( L_1 \), \( L_2 \) is recursive.
Y N If \( L = L_1 \cap L_2 \) and \( L \) is recursive, then one of \( L_1 \), \( L_2 \) is recursive.
Y N If \( L_1 \) is recursive and \( L_2 \) is r.e., then \( L_1 \cap L_2 \) is r.e.
Y N If \( L \) is r.e., then \( \bar{L} \), the complement of \( L \), is r.e.
Problem 4. True/False. Correct = 1, incorrect = -1, no answer = 0.

Y N All recursive languages are r.e.
Y N The complement of a r.e. language is r.e.
Y N The complement of a recursive language is recursive.
Y N If M does not halt on some inputs then L(M) is not recursive.
Y N If L and L̄ are r.e., then L is recursive.
Y N If L, L' are context-free then L ∩ L' is recursive.
Y N If L is recursive and L' is regular then L ∩ L' is regular.
Y N If L is not recursive then L is infinite.
Y N If all words of L can be enumerated by a Turing Machine then L is r.e.
Y N If a TM M is deterministic then L(M) is recursive.

Problem 5. True/False/Open. Correct = 1, incorrect = -1, no answer = 0.

T F O CFL ⊆ P
T F O P = NP
T F O L = PSPACE
T F O NODE COVER \leqL SATISFIABILITY
T F O If a TM M does not halt on some inputs then L(M) is not recursive
T F O If L₁, L₂ are r.e. then so is L₁ ∩ L₂
T F O PSPACE ≠ class of recursive languages
T F O If TSP ∈ P then P = NP
T F O TSP \leqL QSAT (Quantified boolean expressions)
T F O QSAT \leqL TSP

Problem 6. Briefly define or state.

1. The Universal Turing Machine
2. Savitch’s Theorem
3. The Satisfiability Problem
4. NP and NP-completeness
5. Pseudopolynomial algorithm

Problem 7. For each of the problems stated below, give the smallest complexity class among L, NL, P, NP, coNP, NP ∩ coNP, PSPACE, recursive, r.e, to which the problem is known to belong. (If it’s not even r.e., write “non r.e.”.)

REACHABILITY: Give a directed graph G and two vertices s, t, is there a directed path from s to t in G?
CIRCUIT VALUE: Give a boolean circuit, C and the input values for C, is the output value 1?
PRIMALITY: Given an integer n, is n prime?
RELATIVE PRIMALITY: Given two integers n, m, are they relatively prime? In other words, is their greatest common divisor equal 1?
SAT-CNF: Given a boolean expression α in conjunctive normal form (ands of ors of literals), is α satisfiable?
SAT-DNF: Given a boolean expression $\alpha$ in disjunctive normal form (ors of ands of literals), is $\alpha$ satisfiable?

CFL EMPTINESS: Given a context-free grammar $G$, is $L(G)$ empty?

CFL EVERYTHINGNESS: Given a context-free grammar $G$ over alphabet $\Sigma$, is $L(G) = \Sigma^*$?

DCFL EVERYTHINGNESS: Given a deterministic pushdown automaton $M$ over alphabet $\Sigma$, is $L(M) = \Sigma^*$?

GEOGRAPHY: Given a directed graph $G$ and an initial node $s \in G$, is there a winning strategy in GEOGRAPHY on $G$ for player I?

**Problem 8.** Define or state.

(a) Recursively enumerable language.

(b) Church’s thesis.

(c) The linear speed-up theorem.

(d) Nondeterministic Turing machine.

(e) Relation between nondeterministic and deterministic time complexity classes.

**Problem 9.** For each of the classes below name problems complete in this class.

**NL** (1 problem)

1.

**P** (2 problems)

1.

2.

**NP** (4 problems)

1.

2.

3.

4.

**Problem 10.** True/False. Correct = 1, incorrect = -1, no answer = 0. If $L, L'$ are languages (or decision problems), notation $L \leq_{L} L'$ means that $L$ reduces to $L'$ in logarithmic space.

T F If $L_1 \leq_{L} L_2$ and $L_2 \leq_{L} L_3$ then $L_1 \leq_{L} L_3$.

T F If $L_1 \leq_{L} L_2$ and $L_1$ is in P then $L_2$ is in P.

T F If $L_1 \leq_{L} L_2$ and $L_2$ is in NL then $L_1$ is in NP.

T F If $L_1 \leq_{L} L_2$ and $L_2$ is NL-complete then $L_1$ is in NL-complete.

T F If $L_1 \leq_{L} L_2$ and $L_2$ is P-complete then $L_1$ is in P.

T F If $L_1 \leq_{L} L_2$, $L_1$ is NP-complete and $L_2$ is in P, then $P = NP$.

T F If $L$ is in NL then $L \leq_{L} \text{CIRCUIT\_VALUE}$.

T F If $L$ is in P then \text{CIRCUIT\_VALUE} $\leq_{L} L$.

T F \text{CIRCUIT\_VALUE} $\leq_{L} \text{SAT}$.
Problem 11. For each of the decision problems below, give the smallest complexity class of L, NL, P, NP, coNP or PSPACE, to which (you think) this problem belongs. (Assume that all these classes are different.)

1. Given a graph $G$ and a number $K$, does $G$ have a clique of size $\geq K$?
2. Given a graph $G$ and a number $K$, does $G$ have a clique of size $\leq K$?
3. Given a graph $G$ and a number $K$, do all cliques in $G$ have size $\leq K$?
4. Given a graph $G$ and a number $K$, does $G$ have a maximal clique of size $\leq K$?

Problem 12. True/False/Open. Correct = 1, incorrect = -1, no answer = 0.

- $T$ F O VERTEX_COVER $\leq_L$ 3CNF
- $T$ F O VERTEX_COVER $\in$ P
- $T$ F O If NP $\neq$ coNP then VERTEX_COVER $\notin$ coNP
- $T$ F O PRIMES $\in$ NP $\cap$ coNP
- $T$ F O PRIMES $\in$ P

Problem 13. Tell which of the following statements are true, false or open. Correct = 1, incorrect = -1, no answer = 0.

- $T$ F O If $L$ is r.e. then so is its complement
- $T$ F O If $L_1, L_2$ are r.e. then so is $L_1 \cap L_2$
- $T$ F O If $L_1 \cup L_2$ is r.e. then so are $L_1, L_2$
- $T$ F O If $L$ and its complement are r.e. then $L$ is recursive
- $T$ F O PCP for 2-letter alphabets is undecidable
- $T$ F O QBF $\in$ L
- $T$ F O QBF $\in$ coNP
- $T$ F O TSP $\leq$ HAMILTONIAN_CYCLE
- $T$ F O TSP $\in$ PSPACE
- $T$ F O PRIMES $\in$ P

Problem 14. For each of the problems below give the smallest class of L, NL, P, NP, coNP, NP $\cap$ coNP, PSPACE, R (recursive), ER (recursively enumerable) to which it belongs. Write “nonRE” if the problem is not recursively enumerable.

1. Given a graph $G$, and a constant $K$, does $G$ have a vertex cover of size $< K$?
2. Given a weighted graph $G$, and a number $K$, does $G$ have spanning tree of weight $\leq K$?
3. Given a Turing machine $M$, is $L(M) \neq \emptyset$?
4. Given two words $x, y$, is $x$ a substring of $y$?
5. Given a polynomial $p(x_1, \ldots, x_k)$ with integer coefficients, does $p(x_1, \ldots, x_k) = 0$ have an integer solution?
6. Given a context-free grammar $G$, is $L(G) = \emptyset$?

1 Or, more precisely, to which it is known to belong.
(7) Given an integer \( x \), is \( x \) composite?

(8) Given a nondeterministic finite automaton \( M \) and a string \( w \), does \( M \) accept \( w \)?

(9) Given a Turing machine \( M \), is \( L(M) \) regular?

(10) Given a graph \( G \) and a number \( K \), is the maximum clique size of \( G \) equal to \( K \)?

**Problem 15.** Define or state.

Undecidability.

Rice’s Theorem.

Savitch’s Theorem.

NP-completeness.

The Satisfiability Problem and Cook’s Theorem.

**Problem 16.** Assuming that all classes \( L, NL, P, NP, coNP, PSPACE, R, RE \) are different, tell which of the following statements are true, false or open. Correct = 1, incorrect = -1, no answer = 0.

\[
\begin{array}{ccc}
T & F & O \\
L \subseteq NP & \text{REACHABILITY is in P} & \text{PRIMES} \leq_L \text{VERTEX}\_\text{COVER} \\
\text{VERTEX}\_\text{COVER} \leq_L \text{PRIMES} & \text{CNF is not in P} & \text{CFL} \subseteq P \\
\text{if } L \text{ is P-complete the so is its complement} & \text{if } L \text{ is NP-complete the so is its complement} & \text{PCP is in PSPACE} \\
\text{SAT for formulas in disjunctive normal form (\lor's of \land's) is NP-complete.} & \text{The problem whether a boolean formula is a tautology (true for all 0/1 assignments) is NP-complete} & \\
\end{array}
\]

**Problem 17.** True or false. In all of the questions regarding complexity theory assume that \( P \neq NP, NP \neq coNP, DLOG \neq NLOG, P \neq NLOG, \) etc.

**YES**  **NO**  **OPEN**

There is a finite automaton that accepts the language \( L = \{a^i b^j \mid i^2 + j^2 \geq 637\} \).

If \( G \) is not context-free then \( L(G) \) is not context-free.

REACHABILITY \( \leq_P \) VC (vertex cover).

VC \( \leq_P \) REACHABILITY.

If \( L \in P \) and \( L' \subseteq L \) then \( L' \) cannot be NP-complete.

**Problem 18.** For each of the following statements, tell whether it is true, false or open.

**YES**  **NO**  **OPEN**

If \( L_1 \in NP \) and \( L_2 \in coNP \) then \( L_1 \cap L_2 \in PSPACE \).
YES NO OPEN P = NLOG.
YES NO OPEN The problem “For a dfa $M$, is $L(M) = \Sigma^*$” is decidable.
YES NO OPEN The problem “For a cfg $G$, is $L(G) = \emptyset$” is decidable.
YES NO OPEN The problem “For a cfg $G$, is $L(G) = \Sigma^*$” is decidable.

Problem 19. True or false. In all of the questions regarding complexity theory assume that $P \neq NP$, $NP \neq coNP$, $DLOG \neq NLOG$ and $P \neq NLOG$.

YES NO The language $L = \{a^ib^j \mid i^2 - j^2 \geq 637\}$ is regular.
YES NO The language $L = \{a^ib^j \mid i^2 - j^2 \geq 637\}$ is context-free.
YES NO If $L$ is context-free then $L' = \{ww^R \mid w \in L\}$ is also context-free.
YES NO PSA$^2 \leq_P$ VERTEX COVER.
YES NO VERTEX COVER $\leq_P$ PSA.
YES NO Let $L_1, L_2 \subseteq \{a, b\}^*$. If $L_1, L_2$ are r.e. then $L = aL_1 \cup bL_2$ is r.e.
YES NO Let $L_1, L_2 \subseteq \{a, b\}^*$. If $L = aL_1 \cup bL_2$ is r.e. then $L_1, L_2$ are r.e.
YES NO If $L_1 \in NP$ and $L_2 \in coNP$ then $L_1 - L_2 \in NP$.
YES NO If $L_1 \in NP$ and $L_2 \in coNP$ then $L_1 - L_2 \in coNP$.
YES NO If $L_1 \in NP$ and $L_2 \in coNP$ then $L_1 - L_2 \in PSPACE$.

Problem 20. True, false or open. Incorrect answers will be given negative credits.

<table>
<thead>
<tr>
<th>TRUE</th>
<th>FALSE</th>
<th>OPEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRUE</td>
<td>FALSE</td>
<td>OPEN</td>
</tr>
<tr>
<td>TRUE</td>
<td>FALSE</td>
<td>OPEN</td>
</tr>
<tr>
<td>TRUE</td>
<td>FALSE</td>
<td>OPEN</td>
</tr>
<tr>
<td>TRUE</td>
<td>FALSE</td>
<td>OPEN</td>
</tr>
<tr>
<td>TRUE</td>
<td>FALSE</td>
<td>OPEN</td>
</tr>
<tr>
<td>TRUE</td>
<td>FALSE</td>
<td>OPEN</td>
</tr>
<tr>
<td>TRUE</td>
<td>FALSE</td>
<td>OPEN</td>
</tr>
<tr>
<td>TRUE</td>
<td>FALSE</td>
<td>OPEN</td>
</tr>
</tbody>
</table>

2Psa = Path Systems Accessibility
**Problem 21.** For each of the decision problems given below, place it in the smallest class of \(\text{PSPACE}, \text{coNP}, \text{NP}, \text{P}, \text{NLOG}, \text{DLOG}, \text{RECURSIVE}, \text{R.E.}\), to which you believe it belongs.

1. Given three integers \(x, y, z\) is \(z = xy\)?
2. Given two graphs \(G_1, G_2\) is \(G_1\) a subgraph of \(G_2\)?
3. Given two graphs \(G_1, G_2\) is \(G_1 \neq G_2\)?
4. Given a network \(D\), source \(s\) and sink \(t\) of \(D\), and an integer \(K\), is there a flow from \(s\) to \(t\) in \(D\) with value \(\geq K\)?
5. Given a quantified boolean formula \(\alpha\) without free variables, is \(\alpha\) false?
6. Given a graph \(G\), does \(G\) have an eulerian cycle (a cycle that traverses each edge exactly once)?
7. Given a digraph \(D\), and two vertices \(s, t\) of \(D\), is there a path from \(s\) to \(t\) of length at most \(\log n\)?
8. Given a deterministic finite automaton \(A\), is \(A\) minimal?
9. Given a context-free grammar \(G\), is \(L(G) = \emptyset\)?
10. Given a context-free grammar \(G\), is \(L(G) \neq \Sigma^*\)?

**Problem 22.** True/False. Correct answer = 1, incorrect = -1, no answer = 0.

0. T F \(\text{SPACE}(\log^2 n) \subseteq \text{TIME}(2^n)\).
1. T F The Halting Problem is undecidable.
2. T F If \(L_1, L_2 \in \text{NP}\) then \(L_1 \cap L_2 \in \text{NP}\).
3. T F If \(L_1, L_2 \in \text{CFL}\) then \(L_1 \cap L_2 \in \text{CFL}\).
4. T F The Post Correspondence Problems is decidable.
5. T F The Clique Problem is decidable.
6. T F \(\text{NP} \cup \text{coNP} \subseteq \text{PSPACE}\).
7. T F Either \(\text{DLOG} \neq \text{P}\) or \(\text{P} \neq \text{PSPACE}\).
8. T F If \(L \in \text{P}\) then \(\bar{L} \in \text{P}\). (\(\bar{L}\) is the complement of \(L\).)
9. T F If \(L \in \text{NP}\) then \(\bar{L} \in \text{NP}\).

**Problem 23.** True/False/Open. Correct = 1, incorrect = -1, no answer = 0.

\[
\begin{array}{ccc}
\text{T} & \text{F} & \text{O} \\
\text{NP} \cap \text{coNP} = \text{P} & \text{SATISFIABILITY} \in \text{P} & \text{If } L \text{ is regular then } L \in \text{TIME}(n) \\
\text{If } L_1, L_2 \text{ are context-free then } L_1 \cap L_2 \text{ is context free} & \text{If } L_1, L_2 \text{ are r.e. then } L_1 \cap L_2 \text{ is r.e.} & \text{If } L \text{ is finite then } L \text{ is regular} \\
\text{If } L = \text{P then } \text{P} \neq \text{PSPACE} & \text{REACHABILITY} \text{ is NL-complete} & \text{CIRCUIT VALUE} \text{ is NL-complete} \\
\text{PRIMALITY} \in \text{P} & \end{array}
\]
Note: L = LOGSPACE, NL = NLOGSPACE.

**Problem 24.** True/False. Correct answer = 1, incorrect = -1, no answer = 0.

0. T F Context-free languages are recursive
1. T F The Halting Problem is decidable.
2. T F The Halting Problem restricted to machines with 5 states is decidable
3. T F If \(L_1, L_2 \in \text{CFL}\) then \(L_1 - L_2 \in \text{CFL}\).
4. T F \(\text{NSPACE}(n \log n) \subseteq \text{SPACE}(n^{5/2})\)
5. T F If there exists a polynomial-time algorithm for 3CNF then \(P = NP\)
6. T F \(\text{CLIQUE} \leq_p \text{3CNF}\)
7. T F \(\text{MINIMUM SPANNING TREE} \leq_p \text{3CNF}\)
8. T F \(\text{PRIMALITY} \in \text{NP}\)
9. T F If \(L_1, L_2 \in \text{NP}\) then \(L_1 \cup L_2 \in \text{NP}\).

**Problem 25.** True/False/Open. Correct = 1, incorrect = -1, no answer = 0.

\[
\begin{array}{ccc}
T & F & O \\
\text{SPACE}(\log^2 n) & \subseteq & P \\
\text{SPACE}(\sqrt{n \log n}) & \subseteq & P \\
\text{CFL} & \subseteq & P \\
\text{If } L_1, L_2 \text{ are context-free then so is } L_1 \cup L_2 \\
\text{If } L_1, L_2 \text{ are in NP, then so is } L_1 \cap L_2 \\
\text{QBF is PSPACE-complete} \\
\text{QBF is in L} \\
\text{QBF is in P} \\
\text{HALTING PROBLEM is in PSPACE} \\
\text{If } L \text{ is in P, then so is its complement}
\end{array}
\]

Note: \(L = \text{LOGSPACE}, NL = \text{NLOGSPACE}, \text{CFL} = \text{context-free languages}, \text{QBF} = \text{“is a given quantified boolean formula true?”}\).

**Problem 26.** True/False/Open. Correct = 1, incorrect = -1, no answer = 0.
Problem 27. What is the relationship, if any, between the following complexity classes.

(a) $\text{DSPACE}(n^5)$ and $\text{NSPACE}(n^2 \log^2 n)$.
(b) $\text{DSPACE}(3^n)$ and $\text{DTIME}(n^n)$.
(c) $\text{DTIME}(n2^n)$ and $\text{DSPACE}(n \log \log n)$.
(d) $\text{NTIME}(3^n)$ and $\text{DTIME}(2^{2n})$.

Problem 28. True/False/Open. Correct =1, incorrect = -1, no answer = 0.

(a) $\text{P} = \text{NP}$
(b) If $\text{P} = \text{NP}$ then $\text{coNP} = \text{NP}$.
(c) $\text{NL} = \text{coNL}$
(d) $\text{PCP}$ is decidable
(e) $\text{PCP}$ is recursively enumerable
(f) $\text{TQBF} \in \text{NP}$
(g) $\text{TQBF} \in \text{NL}$
(h) $\text{A_{TM}} \in \text{EXPTIME}$
(i) $\text{PATH} \in \text{SPACE}(\log^2 n)$

The following comment could be helpful in one problem: In class we used polynomial-time reductions to define NP-completeness and PSPACE-completeness. However, all reductions that we covered in class can actually be done in logarithmic space.

Problem 29. Give a complete definition or statement:

(a) Savitch’s Theorem
(b) Logspace reducibility
(c) Hilbert’s 10th Problem
(d) Undecidability
(e) Generalized Geography