Tutorial (Complexity of Problems)

Exercise 1

The graph G is Eulerian if there exists a cycle that traverses each edge of graph G exactly once. Recall that a connected graph is Eulerian if and only if each of its vertices has an even degree.

1. Write down the associated decision problem.
2. Find a polynomial algorithm that determines if the graph is Eulerian.

Exercise 2

Formulate the following problems as decision problems:

1. Given a graph, determine if there exists a path between two disjoint vertices.
2. Given an unweighted graph, determine the existence of the shortest path between two disjoint vertices.
3. Given a weighted graph, determine the existence of the longest elementary chain. A chain is elementary if it doesn't visit the same vertex twice.
4. Find the smallest nontrivial divisor[1] of a natural number n.

Exercise 3

Are the following problems in NP, in P? Justify your answer.

1. Problem P_1:

   Data: A graph $G = (V, E)$

   Question: Does there exist a cycle of length equal to 4?

2. Problem P_2:

   Data: A graph $G = (V, E)$, two distinct vertices u and v in G, and an integer k.

   Question: Is there a simple path between u and v with a length less than or equal to k?

Exercise 4

Show that the following problems are in the complexity class P:

1. Problem DNF-SAT:

   Input: A propositional formula φ in disjunctive normal form.

   Question: Determine whether there is an assignment of truth values to the propositional variables that makes φ satisfiable.

2. Problem CNF-Taut:

   Input: A propositional formula φ in conjunctive normal form.

   Question: Determine whether φ is always true (regardless of the truth assignments).

Exercise 5

Consider the 2-SAT problem defined as follows:

Data: A set of variables $U = \{x_1, ..., x_n\}$ and a formula $\varphi = c_1 \wedge c_2 \wedge ... \wedge c_p$ with $c_i = y_{i,1} \vee y_{i,2}$ where for all i, j, $y_{i,j}$ is a literal, meaning $y_{i,j}$ is either x_k or $\neg x_k$ for some x_k.

Question: Is there an assignment of truth values to the variables that makes φ true?

1. Show that $u \vee v = (\neg u \Rightarrow v) \wedge (\neg v \Rightarrow u)$.
   From an instance (U, φ) of 2-SAT, construct a directed graph $G_\varphi = (V, E)$ such that:
   - One vertex for each literal in U.
   - One arc for each implication (transforming each clause into two implications).

[1] A nontrivial divisor of a natural number n is a natural number that divides n but is not equal to n or 1 (which are its trivial divisors)

2. Draw the graphs G_{φ_1} and G_{φ_2} for:

$$\varphi_1 = (y \vee \neg z) \wedge (\neg y \vee z) \wedge (y \vee \neg x)$$

$$\varphi_2 = (y \vee z) \wedge (\neg y \vee z) \wedge (\neg z \vee x) \wedge (\neg z \vee \neg x)$$

3. Show that if G_φ has a path from u to v, then it also has a path from $\neg v$ to $\neg u$.
4. Show that there exists a variable u in U such that G_φ contains a cycle from u to $\neg u$ if and only if φ is unsatisfiable.
5. Show that a polynomial-time algorithm can solve the 2-SAT problem.

Exercise 6

Let A and B be two decision problems.
We denote $X \subseteq D$ as the set of positive instances of problem A, and $Y \subseteq D'$ as the set of positive instances of problem B.
A polynomial reduction from A to B is a polynomial-time algorithm (computable function in polynomial time) $f : D \longrightarrow D'$ such that:

$$x \in X \Longleftrightarrow f(x) \in Y$$

We write $A \leq_p B$ when A is polynomially reducible to B. $A \leq_p B$ implies that problem A is therefore easier than problem B.

1. Show that $\leq_p$ is reflexive ($A \leq_p A$).
2. Show that $\leq_p$ is transitive ($A \leq_p B$ and $B \leq_p C \Longrightarrow A \leq_p C$).
3. Show that if $B \in NP$, $A \leq_p B$, and $A \in NPC$, then $B \in NPC$.
4. Show that if $A \leq_p B$ and $B \in P$, then $A \in P$.
5. Show that $P = NP$ if and only if $SAT \in P$.
6. Let A be an NP-complete problem. Show that $P = NP$ if and only if $A \in P$.

Exercise 7

Let the 2-COLORABILITY problem be defined as follows:

Data: An undirected graph $G = (V, E)$.

Question: Does there exist a graph coloring using at most 2 colors such that adjacent vertices do not receive the same color?

- Prove that the 2-COLORABILITY problem is polynomially reducible to 2-SAT. Conclude that it is in P.

Exercise 8

Consider the NAESAT problem defined as follows:

Data: A set of variables $\{x_1, ..., x_n\}$ and a set of clauses $c_1, c_2, ..., c_l$ where $c_i = y_{i,1} \vee ... \vee y_{i,k_i}$ and for all i, j, $y_{i,j}$ is a literal, meaning $y_{i,j}$ is either x_k or $\neg x_k$ for some x_k.

Question: Is there a truth value assignment to the variables $x_i \in \{0, 1\}$ such that each clause contains at least one true literal and at least one false literal (i.e., for each i, there exists j and k such that $y_{i,j} = 1$ and $y_{i,k} = 0$)?

- Prove that this problem is NP-complete by reducing it from the SAT problem.

Exercise 9

A clique in a graph is a set of vertices where each vertex is connected by an edge to all other vertices in the set. Thus, a clique in a graph is a subgraph that is a complete graph.
The CLIQUE problem is defined as follows:

Data: An undirected graph $G(V, E)$ and an integer k.

Question: Does G contain a clique of size k?

- Prove the NP-completeness of this problem by reducing it from the 3-SAT problem, knowing that 3-SAT is NP-complete (in the course, the STABLE problem was used for this proof).

Exercise 10

Consider the VERTEX COVER problem defined as follows:

Data: An undirected graph $G = (V, E)$ and an integer k.

Question: Does there exist a subset $V' \subset V$, with $|V'| = k$, such that every edge in G has at least one endpoint in V'?

- Prove that this problem is NP-complete (use the CLIQUE problem).

Exercise 11

Consider the single-tape Turing machine $M = (Q, q_0, F, \Sigma, \Gamma, \delta)$:

- The set of states Q is $\{q_0, q_1, q_2\}$.
- The initial state is q_0.
- The only final state is state q_2.
- The input alphabet $\Sigma = \{0, 1\}$.
- The tape alphabet is $\Gamma = \{0, 1, \square\}$.
- The set of transitions $\delta = \{(q_0, 0, \square, \rightarrow, q_0), (q_0, 1, \square, \rightarrow, q_1), (q_1, 1, \square, \rightarrow, q_1), (q_1, \square, \square, \rightarrow, q_2)\}$.

1. Represent the Turing machine M graphically.
2. What language does it recognize?
3. Write down the computation of the Turing machine M during its execution on the input 001110.

Exercise 12

1. Build a Turing machine M that accepts the language $\{a^n b^n | n \geq 0\}$.
2. Provide the computation of the machine with the input $aaabbb$.
3. Modify this machine to recognize the language $\{a^n b^{2n} | n \geq 0\}$.

Exercise 13

Build a Turing machine that:

1. Interprets its input as a binary integer n, replaces it with $\lfloor \frac{n}{2} \rfloor$, and halts.
2. Performs binary increment.
3. Performs addition of two unary integers.