Decision Problems

- We will consider only problem with Yes-No answers.

Example | Traveling Salesman Problem (TSP)

1. Given a nonnegative integer edge weighted graph, what is the minimum length cycle that visits each node exactly once?

2. Given a nonnegative integer edge weighted graph and an integer $K$, is there a cycle that visits each node exactly once, with weight at most $K$?
A Boolean formula is called **conjunctive normal form (CNF)** if for the input Boolean variable $x_1, x_2, \ldots, x_n$, it has the following form:

$$x_{out} = (x_1 \lor \neg x_1 \lor x_2) \land (x_3 \lor \neg x_2 \lor x_1) \land (x_2 \lor x_1) \land (x_3)$$

Output Boolean variable

clause

i.e. $\text{CNF} = \text{product-of-sums}$
2-CNF: each clause has two distinct literals.

\[ x_{out} = (x_1 \lor \neg x_1) \land (x_3 \lor x_2) \land (x_2 \lor x_1) \land (x_1 + \overline{x_1}) \land (x_3 + x_2) \land (x_2 + x_1) \]

3-CNF: each clause has three distinct literals.

\[ x_{out} = (x_1 \lor \neg x_1 \lor x_2) \land (x_3 \lor \neg x_2 \lor x_1) \land (x_2 \lor x_1 \lor x_3) \]
Some Problems: \( P \) vs \( NP \)-complete

- **2SAT**: Given a Boolean formula in 2-CNF, is there a satisfying truth assignment to the input variables?
- **3SAT**: Given a Boolean formula in 3-CNF is there a satisfying truth assignment to the input variables?
- **SAT**: Is a given \( n \)-variable Boolean formula in CNF satisfiable? Here, “satisfiable” means “can be made true.”

**Example**: \((x+y)(\overline{x}+z)\) is satisfiable  
\((x+z)(\overline{x})(\overline{z})\) is not satisfiable
Definition: A problem $A$ is in $P$ if for every input $x$ the solution $A(x)$ can be computed in polynomial time $O(poly(length(x)))$. 
A problem $A$ is NP if there exist a polynomial $p$ and a polynomial-time algorithm $V()$ such that $x$ is a YES-input for problem $A$ if and only if there exists a solution $y$, with $\text{length}(y) \leq p(\text{length}(x))$ such that $V(x, y)$ outputs YES.

NP = Non-deterministic Polynomial-time
Intuitively, a problem is in \textbf{NP} if it can be formulated as the problem of whether there is a solution

- They are \textit{small}. In each case the solution would never have to be longer than a polynomial in the length of the input.

- They are \textit{easily checkable}. In each case there is a polynomial algorithm which takes as inputs (the input of the problem) and (the alleged solution), and checks whether the solution is a valid one for this input. In the case of \texttt{3SAT}, the algorithm would just check that the truth assignment indeed satisfies all clauses. In the case of \textit{Hamilton cycle} whether the given closed path indeed visits every node once. And so on.

- Every “yes” input to the problem has at least one solution (possibly many), and each “no” input has none.

- For every yes-input $x$ to $A$, there is a polynomial size piece of \textit{evidence}_x which can be checked in polynomial time that indeed $x$ is a yes-input. This \textit{evidence} (sometimes called a \texttt{certificate} or \texttt{witness} of \texttt{proof}) may be very hard to come up with, but is easy to check.
**THEOREM:** \( P \subseteq NP \)

**Proof:** This is obvious. If you can solve a problem efficiently, there exists short evidence as to what the solution is (both for YES and NO inputs!). Simply, use the execution trace of your solving algorithm as the evidence that the answer is YES or NO depending on what the case may be.
3-COLOR

3-COLOR problem: Given a graph $G = (V, E)$, is there a function $c : V \rightarrow \{1, 2, 3\}$ s.t. $c(x) \neq c(y)$ for all edge $xy \in E$?

Fact: 3-COLOR problem is a decision problem.

Fact: 3-COLOR problem is in NP.

pf: Let $G$ be a YES-input for the 3-COLOR problem.

$G$ is 3-colorable $\Rightarrow \exists c : V \rightarrow \{1, 2, 3\}$

$c(x) \neq c(y)$ for any edge $xy$ of $G$

Such a coloring $c$ can serve as a certificate which can be checked in polynomial time that indeed $G$ is 3-colorable.

For each edge $xy \in E(G)$

do if $c(x) = c(y)$ then STOP and $G$ is not 3-colorable.

Print "G is 3-colorable"
Reductions

- A reduction from decision problems $A$ to $B$ is a polynomial-time algorithm $R$ such that for every input $x$ to problem $A$, we have

$$A(x) = B(R(x))$$

$x$ is a “yes” input of $A \Leftrightarrow R(x)$ is a “yes” input of $B$

- We write

$$A \leq B$$
\[ \{A \leq B\} + \{B \text{ has a poly. time algor.}\} \]
\[ \Rightarrow \{A \text{ has a poly. time algor.}\} \]
Reductions

\{ A \leq B \}

⇒ If we known \( A \) is hard,
    then \( B \) is hard.
A problem $B$ is **NP-hard** if for **every problem** $A$ in NP, $A \leq B$.

**NP-hard** = Non-deterministic Polynomial-time hard
NP-complete

Definition

A problem $B$ is **NP-complete** if it is **NP-hard** and it is contained in **NP**.

Lemma

If $B$ is **NP-complete**, then $B$ is in $P$ if and only if $P=NP$. 
Proving more NP-completeness result

Lemma \[ A \leq B \land B \leq C \implies A \leq C \]

Lemma Let \( C \in \text{NP-complete} \) and \( A \in \text{NP} \). If \( C \leq A \) then \( A \in \text{NP-complete} \).
The first NP-complete problem

Cook's Theorem (1971)

SAT is NP-complete.

- Stephen Arthur Cook, "The complexity of Theorem Proving Procedures"
- Cook received Turing Award in 1982.

Millennium Prize Problem: \( P = NP ? \)
**Thm** 3SAT is NP-Complete.

**Pf:** It suffices to show that \( \text{SAT} \leq \text{3SAT} \). (sketch!)

Let \((x_1 + x_2 + \ldots + x_t)(x_1 + x_2 + \ldots + x_t) \ldots (x_1 + x_2 + \ldots + x_t)\) be an input to SAT.

\[
(x) \equiv (x + a + b)(x + \overline{a} + b)(x + a + \overline{b})(x + \overline{a} + \overline{b})
\]

\[
(x + y) \equiv (x + y + c)(x + y + \overline{c})
\]

\[
(x_1 + x_2 + x_3 + \ldots + x_6) \equiv (x_1 + x_2 + \alpha)(\overline{\alpha} + x_3 + \beta)(\beta + x_4 + r)(\overline{r} + x_5 + x_6)
\]

Note that LHS is satisfiable iff RHS is satisfiable.

**QED**
Observations

Claim: Using three colors \{t, f, a\} to color a graph. Suppose we are given the following graph in which vertices \(x, y\) and \(z\) not all receiving the same color. Then we can properly 3-color the other vertices so that vertex \(T\) receives the color \(t\).
proof:

QED
Suppose vertices $x$, $y$ and $z$ received the same color. When we 3-color the other vertices of the graph we will arrive at the following situation. That vertex $T$ always receive the same color as $\{x, y, z\}$. 

![Diagram of the graph with vertices labeled $x$, $y$, $z$, and $T$. The diagram shows relationships between these vertices, with some vertices colored.]
**3-COLOR**

**Thm** 3-COLOR problem is NP-complete.

**pf:** To show $3SAT \leq 3\text{-COLOR}$. (sketch)

Let $S = (x+y+\bar{z}) (\bar{x}+y+z) (\bar{x}+y+\bar{z})$ be an input of $3SAT$.

Let $G_s$ be an input of $3\text{-COLOR}$, where $G_s \equiv S$.

We claim that $S$ is a "yes" input of $3SAT$ if and only if $G_s$ is a "yes" input of $3\text{-COLOR}$. Why?

QED