Algorithm Design and Analysis
Lecture 20

NP-Completeness

• More NPC problems.
More NPC problems

• How to prove a problem A is NPC?
  • (1) prove it is in NP.
  • (2) find an NPC problem B, prove $B \leq_p A$

• The first NPC problem.
• Circuit Satisfiability Problem.
More NPC problems

• SAT Problem:
Boolean Satisfiability Problem

• A boolean formula:
• 1. $n$ boolean variables: $x_1, \ldots, x_n$
• 2. $m$ boolean connectives: $\land$ (AND), $\lor$ (OR), $\neg$ (NOT), $\rightarrow$ (implication), $\leftrightarrow$ (if and only if).
• 3. parentheses

• E.g. $\Phi = ((x_1 \rightarrow x_2) \lor ((\neg x_1 \leftrightarrow x_3) \lor x_4)) \land \neg x_2$

• Assignment: assigning each variable a value.
• Each assignment is associated with an output.

• Assignment: $x_1 = 0, \ x_2 = 0, \ x_3 = 1, \ x_4 = 1$
• Output: $\Phi = ((x_1 \rightarrow x_2) \lor ((\neg x_1 \leftrightarrow x_3) \lor x_4)) \land \neg x_2 = 0$ (exercise)
Boolean Satisfiability Problem

• A boolean formula:
  • 1. \( n \) boolean variables: \( x_1, \ldots, x_n \)
  • 2. \( m \) boolean connectives: \( \land (\text{AND}), \lor (\text{OR}), \neg (\text{NOT}), \rightarrow (\text{implication}), \leftrightarrow (\text{if and only if}) \).
  • 3. parentheses

• E.g. \( \Phi = ((x_1 \rightarrow x_2) \lor ((\neg x_1 \leftrightarrow x_3) \lor x_4)) \land \neg x_2 \)

• A boolean formula is **satisfiable** if there is an assignment giving output 1.

• Input size: polynomial in \( n \) and \( m \).

• **SAT Problem**: given a boolean formula, is the formula satisfiable?
Boolean Satisfiability Problem

- **SAT Problem**: given a boolean formula, is the formula satisfiable?

- (Theorem) SAT is NP complete.
  - 1. $SAT \in NP$: we can verify it in polynomial time.
  - 2. **SAT is NP-hard**: Circuit Satisfiability $\leq_p SAT$. (Theorem 34.9 textbook)
Boolean Satisfiability Problem

• **SAT Problem**: given a boolean formula, is the formula satisfiable?

• (Theorem) SAT is NP complete.
  1. \( SAT \in NP \): we can verify it in polynomial time.
  2. **SAT is NP-hard**: Circuit Satisfiability \( \leq_p SAT \). (Theorem 34.9 textbook)

• SAT is too complicated. One convenient language is a special SAT problem, 3-SAT.
3-SAT

- Variable: $x_1, \ldots, x_n$
- Literal: $x_1, \ldots, x_n$ and $\neg x_1, \ldots, \neg x_n$

- A boolean formula is in conjunctive normal form ($\text{CNF}$), if it is expressed as an $\text{AND}(\wedge)$ of clauses, each of which is the $\text{OR}(\vee)$ of one or more literals.
- E.g. $(\neg x_3 \lor x_2 \lor x_4 \lor \neg x_1) \wedge (x_2 \lor x_4) \wedge (\neg x_1)$

- A boolean formula is in $k$-conjunctive normal form ($k$-$\text{CNF}$), if it is in CNF and each clause has exactly $k$ distinct literals.

- A boolean formula is in 3-conjunctive normal form ($3$-$\text{CNF}$), if it is in CNF and each clause has exactly 3 distinct literals.
- E.g. $(\neg x_3 \lor x_2 \lor x_4) \wedge (x_2 \lor x_4 \lor \neg x_1) \wedge (\neg x_1 \lor x_2 \lor x_4)$
3-SAT

• A boolean formula is in 3-conjunctive normal form (3-CNF), if it is in CNF and each clause has exactly 3 distinct literals.

• E.g. \((\neg x_3 \lor x_2 \lor x_4) \land (x_2 \lor x_4 \lor \neg x_1) \land (\neg x_1 \lor x_2 \lor x_4)\)

• 3-SAT Problem: given a 3-CNF formula, it is satisfiable?

• 3-SAT is NPC.

  • 3-SAT is not harder than SAT as it is a special case of SAT.
  • 3-SAT is also not easier than SAT.
  • (Theorem) SAT \(\leq_P\) 3-SAT.
    • (Sketch) Every boolean formula can be converted to a 3-CNF with the same satisfiability.

  • S-SAT is in NP.
3-SAT

• A boolean formula is in 3-conjunctive normal form (3-CNF), if it is in CNF and each clause has exactly 3 distinct literals.

• E.g. \((\neg x_3 \lor x_2 \lor x_4) \land (x_2 \lor x_4 \lor \neg x_1) \land (\neg x_1 \lor x_2 \lor x_4)\)

• 3-SAT Problem: given a 3-CNF formula, it is satisfiable?

• 3-SAT is NPC.

  • 3-SAT is not harder than SAT as it is a special case of SAT.
  • 3-SAT is also not easier than SAT.
  • (Theorem) SAT \(\leq_p\) 3-SAT.
    • (Sketch) Every boolean formula can be converted to a 3-CNF with the same satisfiability.

• S-SAT is in NP.
Problems

\[ \leq P \]

Circuit Satisfiability Problem.

\[ \rightarrow \]

SAT

\[ \rightarrow \]

3-SAT
Decision vs Optimization

• Optimization Problem.
• A candidate solution set $S$.
• An objective function $f$.
• Find the $s \in S$ such that $f(s)$ is maximized or minimized.

• E.g.
• Minimum s-t cut problem.
• A candidate solution set $S$: all the possible cuts
• An objective function $f$: the capacity of the cut
• Find a cut such that $f$ is minimized.

• Goal: decide if there is a polynomial algorithm to solve an optimization problem.
Decision vs Optimization

- Optimization Problem.
- A candidate solution set $S$.
- An objective function $f$.
- Find the $s \in S$ such that $f(s)$ is maximized or minimized.

E.g.
- Minimum s-t cut problem.
- A candidate solution set $S$: all the possible cuts
- An objective function $f$: the capacity of the cut
- Find a cut such that $f$ is minimized.

Goal: decide if there is a polynomial algorithm to solve an optimization problem.
Decision vs Optimization

• Optimization Problem.
• A candidate solution set $S$.
• An objective function $f$.
• Find the $s \in S$ such that $f(s)$ is maximized or minimized.

• E.g.
• Minimum s-t cut problem.
• A candidate solution set $S$: all the possible cuts
• An objective function $f$: the capacity of the cut
• Find a cut such that $f$ is minimized.

• Goal: decide if there is a polynomial algorithm to solve an optimization problem.
Decision vs Optimization

- Optimization Problem.
- A candidate solution set $S$.
- An objective function $f$.
- Find the $s \in S$ such that $f(s)$ is maximized or minimized.

- For a maximization problem:
  - Decision Version:
  - Given a number $k$, is there a solution $s$ in $S$, such that $f(s) \geq k$?

- For a minimization problem:
  - Decision Version:
  - Given a number $k$, is there a solution $s$ in $S$, such that $f(s) \leq k$?
Decision vs Optimization

• Optimization Problem.
• A candidate solution set $S$.
• An objective function $f$.
• Find the $s \in S$ such that $f(s)$ is maximized or minimized.

• For a maximization problem:
  • Decision Version:
    • Given a number $k$, is there a solution $s$ in $S$, such that $f(S) \geq k$?

• For a minimization problem:
  • Decision Version:
    • Given a number $k$, is there a solution $s$ in $S$, such that $f(s) \leq k$?
Decision vs Optimization

- Optimization Problem.
- A candidate solution set $S$.
- An objective function $f$.
- Find the $s \in S$ such that $f(s)$ is maximized.

What is the relationship between the decision problem and the optimization problem?

For a maximization problem:
- Decision Version:
  - Given a number $k$, is there a solution $s$ in $S$, such that $f(S) \geq k$?

- If there is an polynomial algorithm to solve the maximization problem, then there is an polynomial algorithm to solve its decision problem for each $k$.
- $\implies$ If for a certain $k$ the decision problem is “hard”, the optimization problem is hard.
Decision vs Optimization

- Optimization Problem.
- A candidate solution set $S$.
- An objective function $f$.
- Find the $s \in S$ such that $f(s)$ is maximized.

- For a maximization problem:
- Decision Version:
  - Given a number $k$, is there a solution $s$ in $S$, such that $f(S) \geq k$?

- If there is a polynomial algorithm to solve the maximization problem, then there is an polynomial algorithm to solve its decision problem for each $k$.
- => If for a certain $k$ the decision problem is “hard”, the optimization problem is hard.

What is the relationship between the decision problem and the optimization problem?
Decision vs Optimization

- Optimization Problem.
- A candidate solution set $S$.
- An objective function $f$.
- Find the $s \in S$ such that $f(s)$ is maximized.

- For a maximization problem:
  - Decision Version:
  - Given a number $k$, is there a solution $s$ in $S$, such that $f(S) \geq k$?

- If there is a polynomial algorithm to solve the maximization problem, then there is a polynomial algorithm to solve its decision problem for each $k$.

- If for a certain $k$ the decision problem is “hard”, the optimization problem is hard.
What is the relationship between the decision problem and the optimization problem?

- Optimization Problem.
- A candidate solution set $S$.
- An objective function $f$.
- Find the $s \in S$ such that $f(s)$ is maximized.

For a maximization problem:
- Decision Version:
  - Given a number $k$, is there a solution $s$ in $S$, such that $f(S) \geq k$?

- If there is a polynomial algorithm to solve the maximization problem, then there is a polynomial algorithm to solve its decision problem for each $k$.
- $\Rightarrow$ If for a certain $k$ the decision problem is “hard”, the optimization problem is hard.
Independent Set

• Given a graph we say a set of nodes $S \subseteq V$ is independent if no two nodes in $S$ are joined by an edge.

• The smallest independent set is trivial. How to find the largest one?

• (Independent Set) Given a graph $G$, find the size of the largest independent set. [Optimization version.]

• (Independent Set) Given a graph $G$ and a number $k$, determine if there is an independent set of size at least $k$. [Decision version.]
Independent Set

• Given a graph we say a set of nodes $S \subseteq V$ is independent if no two nodes in $S$ are joined by an edge.

• The smallest independent set is trivial. How to find the largest one?

• (Independent Set) Given a graph $G$, find the size of the largest independent set. [Optimization version]

• (Independent Set) Given a graph $G$ and a number $k$, determine if there is an independent set of size at least $k$. [Decision version]
Independent Set

- Given a graph we say a set of nodes $S \subseteq V$ is independent if no two nodes in $S$ are joined by an edge.

- Enumerate all the solutions?

- Randomly add a feasible node to the current set?
Independent Set

• Given a graph we say a set of nodes $S \subseteq V$ is independent if no two nodes in $S$ are joined by an edge.

• Enumerate all the solutions?

• Randomly add a feasible node to the current set?
Independent Set

- Given a graph we say a set of nodes $S \subseteq V$ is independent if no two nodes in $S$ are joined by an edge.

- Enumerate all the solutions?

- Randomly add a feasible node to the current set?
Independent Set

• Given a graph we say a set of nodes $S \subseteq V$ is independent if no two nodes in $S$ are joined by an edge.

• Enumerate all the solutions?

• Randomly add a feasible node to the current set?
Independent Set

- Given a graph we say a set of nodes $S \subseteq V$ is independent if no two nodes in $S$ are joined by an edge.
Independent Set

• Given a graph we say a set of nodes $S \subseteq V$ is independent if no two nodes in $S$ are joined by an edge.

• Enumerate all the solutions?

• Randomly add a feasible node to the current set?

• Smartly select the node?
  • We do not know if we can find the optimal in polynomial.
Independent Set

• (Independent Set) Given a graph $G$, find the size of the largest independent set. [Optimization version]

• (Independent Set) Given a graph $G$ and a number $k$, determine if there is an independent set of size at least $k$. [Decision version]

• Comparing the above two problems in terms of polynomial computability:
  • If the optimization version is polynomial solvable, the decision version is polynomial solvable.
  • If the decision version is polynomial solvable for each $k$, the optimization version is polynomial time solvable.
Independent Set

• (Independent Set) Given a graph $G$, find the size of the largest independent set. [Optimization version]

• (Independent Set) Given a graph $G$ and a number $k$, determine if there is an independent set of size at least $k$. [Decision version]

• Comparing the above two problems in terms of polynomial computability:
  • If the optimization version is polynomial solvable, the decision version is polynomial time solvable.
  • If the decision version is polynomial solvable for each $k$, the optimization version is polynomial solvable.
Independent Set

• (Independent Set) Given a graph $G$ and a number $k$, determine if there is an independent set of size at least $k$. [Decision version]

• The decision version of the independent set problem is NP-complete.

• It is in NP.

• It is NP-hard. We prove this by showing $3$-SAT $\leq_P$ Independent Set.
Independent Set

- 3-SAT \( \leq_p \) Independent Set.
- Given an instance of 3-SAT with \( n \) variables \( \{x_1, ..., x_n\} \) and \( k \) clauses \( C_1, ..., C_k \).
- \((l_1^1 \lor l_2^1 \lor l_3^1) \land (l_1^2 \lor l_2^2 \lor l_3^2) \land (l_1^3 \lor l_2^3 \lor l_3^3) \land ... \land (l_1^k \lor l_2^k \lor l_3^k)\)
- \(l_i^j \in \{x_1, ..., x_n, \neg x_1, ..., \neg x_n\}\). Two literals \( a_1, a_2 \) are **conflict** if \( a_1 = \neg a_2 \).
Independent Set

• $3\text{-SAT} \leq_p$ Independent Set.
• Given an instance of 3-SAT with $n$ variables $\{x_1, \ldots, x_n\}$ and $k$ clauses $C_1, \ldots, C_k$.
• $(l_1^1 \lor l_2^1 \lor l_3^1) \land (l_1^2 \lor l_2^2 \lor l_3^2) \land (l_1^3 \lor l_2^3 \lor l_3^3) \land \ldots \land (l_k^1 \lor l_k^2 \lor l_k^3)$
• $l_i^j \in \{x_1, \ldots, x_n, \neg x_1, \ldots, \neg x_n\}$. Two literals $a_1, a_2$ are conflict if $a_1 = \neg a_2$.
• We construct an instance of independent set as follows.
  • Construct a graph $G = (V, E)$ with $3k$ nodes grouped into $k$ triangles. Each triangle corresponds to a clause. For the $i$-th triangle, we label the nodes as $l_i^1, l_i^2, l_i^3$, corresponding to the literals in the $i$-th clause.
Independent Set

- $3$-SAT $\leq_p$ Independent Set.

- Given an instance of $3$-SAT with $n$ variables $\{x_1, \ldots, x_n\}$ and $k$ clauses $C_1, \ldots, C_k$.

- $l_1^1 \lor l_2^1 \lor l_3^1 \land (l_1^2 \lor l_2^2 \lor l_3^2) \land (l_1^3 \lor l_2^3 \lor l_3^3) \land \cdots \land (l_1^k \lor l_2^k \lor l_3^k)$

- $l_i^j \in \{x_1, \ldots, x_n, \neg x_1, \ldots, \neg x_n\}$. Two literals $a_1, a_2$ are conflict if $a_1 = \neg a_2$.

- We construct an instance of independent set as follows.
  - Construct a graph $G = (V, E)$ with $3k$ nodes grouped into $k$ triangles. Each triangle corresponds to a clause. For the $i$-th triangle, we label the nodes as $l_1^i, l_2^i, l_3^i$, corresponding to the literals in the $i$-th clause.
  - For each pair of nodes, add an edge between them if they are conflict literals.

![Diagram of a graph with labeled nodes and edges indicating conflict literals](image-url)
Independent Set

- $3\text{-SAT} \leq_p \text{Independent Set}.$
- $(x_1 \lor x_2 \lor x_4) \land (\neg x_1 \lor x_3 \lor \neg x_2)$
Independent Set

• $3$-SAT $\leq_P$ Independent Set.
• $(x_1 \lor x_2 \lor x_4) \land (\neg x_1 \lor x_3 \lor \neg x_2)$
Independent Set

- $3$-SAT $\leq_P$ Independent Set.
- $(x_1 \lor x_2 \lor x_4) \land (\neg x_1 \lor x_3 \lor \neg x_2)$
Independent Set

- $3$-SAT $\leq_p$ Independent Set.
- Given an instance of $3$-SAT with $n$ variables $\{x_1, \ldots, x_n\}$ and $k$ clauses $C_1, \ldots, C_k$.
- $(l_1^1 \lor l_1^2 \lor l_1^3) \land (l_2^1 \lor l_2^2 \lor l_2^3) \land (l_3^1 \lor l_3^2 \lor l_3^3) \land \cdots \land (l_k^1 \lor l_k^2 \lor l_k^3)$
- $l_i^j \in \{x_1, \ldots, x_n, \neg x_1, \ldots, \neg x_n\}$. Two literals $a_1, a_2$ are conflict if $a_1 = \neg a_2$.
- We construct an instance of independent set as follows.
  - Construct a graph $G = (V, E)$ with $3k$ nodes grouped into $k$ triangles. Each triangle corresponds to a clause. For the $i$-th triangle, we label the nodes as $l_i^1, l_i^2, l_i^3$, corresponding to the literals in the $i$-th clause.
  - For each pair of nodes, add an edge between them if they are conflict literals.

The $3$-SAT instance is satisfiable iff the constructed graph has an independent of size $k$. 

![Graph diagram with labeled nodes and edges]
Independent Set

• Part 1. Consider an assignment $A$ of $x_1, \ldots, x_n$ with output 1.

\[(l^1_1 \lor l^2_1 \lor l^3_1) \land (l^1_2 \lor l^2_2 \lor l^3_2) \land (l^1_3 \lor l^2_3 \lor l^3_3)\]
Independent Set

- Part 1. Consider an assignment $A$ of $x_1, \ldots, x_n$ with output 1.
- In each clause, at least one literal has value 1. For each clause, arbitrarily select one literal (node) with value 1, and we have a set of $k$ nodes.

\[(l^1_1 \lor l^2_1 \lor l^3_1) \land (l^1_2 \lor l^2_2 \lor l^3_2) \land (l^1_3 \lor l^2_3 \lor l^3_3)\]
Independent Set

• Part 1. Consider an assignment $A$ of $x_1, \ldots, x_n$ with output 1.

• In each clause, at least one literal has value 1. For each clause, arbitrarily select one literal (node) with value 1, and we have a set of $k$ nodes. We claim this set must be an independent set. That is, there is no edge between any pair of the nodes in this set.

\[(l_1^1 \lor l_1^2 \lor l_1^3) \land (l_2^1 \lor l_2^2 \lor l_2^3) \land (l_3^1 \lor l_3^2 \lor l_3^3)\]
Independent Set

• Part 1. Consider an assignment \( A \) of \( x_1, \ldots, x_n \) with output 1.

• In each clause, at least one literal has value 1. For each clause, arbitrarily select one literal (node) with value 1, and we have a set of \( k \) nodes. We claim this set must be an independent set. That is, there is no edge between any pair of the nodes in this set.

• First, for the pair of the nodes within the same triangle, ...

\[
(l_1^1 \lor l_1^2 \lor l_1^3) \land (l_2^1 \lor l_2^2 \lor l_2^3) \land (l_3^1 \lor l_3^2 \lor l_3^3)
\]
Independent Set

• Part 1. Consider an assignment $A$ of $x_1, \ldots, x_n$ with output 1.

• In each clause, at least one literal has value 1. For each clause, arbitrarily select one literal (node) with value 1, and we have a set of $k$ nodes. We claim this set must be an independent set. That is, there is no edge between any pair of the nodes in this set.

• First, for the pair of the nodes within the same triangle, ...

• Second, for any two nodes crossing triangles, ...

\[(l^1_1 \lor l^2_1 \lor l^3_1) \land (l^1_2 \lor l^2_2 \lor l^3_2) \land (l^1_3 \lor l^2_3 \lor l^3_3)\]
Independent Set

- Part 2. Suppose $G$ has an independent set $S$ of size $k$. 

\[(l_1^1 \lor l_2^1 \lor l_3^1) \land (l_1^2 \lor l_2^2 \lor l_3^2) \land (l_1^3 \lor l_2^3 \lor l_3^3)\]
Part 2. Suppose $G$ has an independent set $S$ of size $k$.

Consider the assignment $A$ where a variable $x$ has value 1 iff there is a literal $l^i_j \in S$ with $l^i_j = x$. We prove $A$ is feasible and $A$ is a true assignment.
Independent Set

- 3-SAT $\leq_P$ Independent Set.
- $(x_1 \lor x_2 \lor x_4) \land (\neg x_1 \lor x_3 \lor \neg x_2)$

Independent set
\{x_2, x_3\}

Assignment
\[
\begin{align*}
x_1 &= 0 \\
x_2 &= 1 \\
x_3 &= 1 \\
x_4 &= 0 \\
x_5 &= 0
\end{align*}
\]
Independent Set

• Part 2. Suppose \( G \) has an independent set \( S \) of size \( k \).

• Consider the assignment \( A \) where a variable \( x \) has value \( 1 \) iff there is a literal \( l_i^j \in S \) with \( l_i^j = x \). We prove \( A \) is feasible and \( A \) is a true assignment.

• \( A \) is feasible iff for each variable \( x \), \( x \) will be not be set as 1 and 0 simultaneously. That is, \( S \) cannot have two literals with \( x \) and \( \neg x \), respectively.
Independent Set

• Part 2. Suppose $G$ has an independent set $S$ of size $k$.

• Consider the assignment $A$ where a variable $x$ has value 1 iff there is a literal $l_i^j \in S$ with $l_i^j = x$. We prove $A$ is feasible and $A$ is a true assignment.

• $A$ is feasible iff for each variable $x$, $x$ will be not be set as 1 and 0 simultaneously. That is, $S$ cannot have two literals with $x$ and $\neg x$, respectively. $A$ must be feasible, because for each pair of nodes $x$ and $\neg x$, there is an edge in the graph between them and therefore they cannot both appear in an independent set.
Independent Set

• Part 2. Suppose $G$ has an independent set $S$ of size $k$.

• Consider the assignment $A$ where a variable $x$ has value 1 iff there is a literal $l_i^j \in S$ with $l_i^j = x$. We prove $A$ is feasible and $A$ is a true assignment.

• $A$ is feasible iff for each variable $x$, $x$ will not be set as 1 and 0 simultaneously. That is, $S$ cannot have two literals with $x$ and $\neg x$, respectively. $A$ must be feasible, because for each pair of nodes $x$ and $\neg x$, there is an edge in the graph between them and therefore they cannot both appear in an independent set.

• Since $|S| = k$ and each triangle has at most one node in $S$, it must be that each triangle has exactly one node in $S$. Therefore, for each clause, there is a literal with value 1. So $A$ is a true assignment and the formula is satisfiable.

\[(l_1^1 \lor l_2^2 \lor l_3^3) \land (l_1^1 \lor l_2^2 \lor l_3^3) \land (l_1^1 \lor l_2^2 \lor l_3^3)\]
Independent Set

• (Independent Set) Given a graph $G$ and a number $k$, determine if there is an independent set of size at least $k$.[Decision version.]

• The decision version of the independent set problem is NP-complete.
• It is in NP.
• It is N-hard. We prove this by showing 3-SAT $\leq_P$ Independent Set.

• (Theorem) Independent Set is NP-complete.
Independent Set

\[ \leq_P \]

Circuit Satisfiability Problem.

\[ \downarrow \]

SAT

\[ \downarrow \]

3-SAT

\[ \rightarrow \]

Independent Set
Vertex Cover

• Given a graph, a set of nodes $S \subseteq V$ is a vertex cover if every edge has least one endpoint in $S$.

• Vertices cover edges.
Vertex Cover

• Given a graph, a set of nodes $S \subseteq V$ is a vertex cover if every edge has least one endpoint in $S$.

• Vertices cover edges.

• Example:
Vertex Cover

• Given a graph, a set of nodes $S \subseteq V$ is a vertex cover if every edge has at least one endpoint in $S$.

• Vertices cover edges.

• The largest vertex is trivial. How to find the smallest one?

• Example:
Vertex Cover

- Given a graph, a set of nodes $S \subseteq V$ is a vertex cover if every edge has least one endpoint in $S$.
- Vertices cover edges.

- The largest vertex is trivial. How to find the smallest one?

- (Vertex Cover) Given a graph $G$ and a number $k$, does $G$ contain a vertex cover of size at most $k$?

- Example:
Vertex Cover

• (Vertex Cover) A set of nodes $S \subseteq V$ is a vertex cover if every edge has least one endpoint in $S$.

• (Independent Set) A set of nodes $S \subseteq V$ is independent if no two nodes in $S$ are joined by an edge.
Vertex Cover

- (Vertex Cover) A set of nodes $S \subseteq V$ is a vertex cover if every edge has least one endpoint in $S$.

- (Independent Set) A set of nodes $S \subseteq V$ is independent if no two nodes in $S$ are joined by an edge.

- Fact. If $S$ is an independent set if and only if its complement $V \setminus S$ is a vertex cover.
Vertex Cover

• (Vertex Cover) A set of nodes $S \subseteq V$ is a vertex cover if every edge has at least one endpoint in $S$.

• (Independent Set) A set of nodes $S \subseteq V$ is independent if no two nodes in $S$ are joined by an edge.

• Fact. If $S$ is an independent set if and only if its complement $V \setminus S$ is a vertex cover.

• Example:
Vertex Cover

- (Vertex Cover) A set of nodes $S \subseteq V$ is a vertex cover if every edge has at least one endpoint in $S$.

- (Independent Set) A set of nodes $S \subseteq V$ is independent if no two nodes in $S$ are joined by an edge.

- Fact. If $S$ is an independent set if and only if its complement $V \setminus S$ is a vertex cover.

- Example:
Vertex Cover

• (Vertex Cover) A set of nodes $S \subseteq V$ is a vertex cover if every edge has at least one endpoint in $S$.

• (Independent Set) A set of nodes $S \subseteq V$ is independent if no two nodes in $S$ are joined by an edge.

• Fact. If $S$ is an independent set if and only if its complement $V \setminus S$ is a vertex cover.

• Suppose $S$ is an independent set and $V \setminus S$ is not vertex cover. Then there is an edge $(u, v)$ such that both $u$ and $v$ are not in $V \setminus S$, which means both of $u$ and $v$ are in $S$, contradicting that $S$ is an independent set.

• Suppose $S$ is a vertex cover and $V \setminus S$ is not an independent set. Then there is an edge $(u, v)$ such that both $u$ and $v$ are in $V \setminus S$, which means both $u$ and $v$ are not in $S$, contradicting that $S$ is a vertex cover.
Vertex Cover

• (Vertex Cover) A set of nodes $S \subseteq V$ is a vertex cover if every edge has least one endpoint in $S$.

• (Independent Set) A set of nodes $S \subseteq V$ is independent if no two nodes in $S$ are joined by an edge

• Fact. If $S$ is an independent set if and only if its complement $V \setminus S$ is a vertex cover.

• Suppose $S$ is an independent set and $V \setminus S$ is not vertex cover. Then there is an edge $(u, v)$ such that both $u$ and $v$ are not in $V \setminus S$, which means both of $u$ and $v$ and in $S$, contradicting that $S$ is an independent set.

• Suppose $S$ is a vertex cover and $V \setminus S$ is not an independent set. Then there is an edge $(u, v)$ such that both $u$ and $v$ are in $V \setminus S$, which means both $u$ and $v$ are not in $S$, contradicting that $S$ is a vertex cover.
Vertex Cover

• Fact. If $S$ is an independent set if and only if its complement $V \setminus S$ is a vertex cover.

• (Vertex Cover) Given a graph $G$ and a number $k$, does $G$ contain a vertex cover of size at most $k$.

• (Independent Set) Given a graph $G$ and a number $k$, determine if there is an independent set of size at least $k$.

• **Vertex Cover $\leq_p$ Independent Set.** Given an instance $(G, k)$ of the vertex cover problem, consider the instance $(G, n - k)$ of the Independent Set problem.

• **Independent Set $\leq_p$ Vertex Cover.** Given an instance $(G, k)$ of the Independent Set problem, consider the instance $(G, n - k)$ of the Vertex Cover.
Problems

\[ \leq_P \]

Circuit Satisfiability Problem.

SAT

3-SAT

Independent Set

Vertex Cover
Set Cover
Set Cover

The pictures belong to their creators.
Set Cover

- There are some bags each of which has some cards.

The pictures belong to their creators.
• There are some bags each of which has some cards.
• Different bags may have the same card. And there are totally $n$ different cards.
There are some bags each of which has some cards.
Different bags may have the same card. And there are totally \( n \) different cards.

You want to buy some bags, as few as possible, so that you will have all the different cards.

The pictures belong to their creators.
Set Cover

- There are some bags each of which has some cards.
- Different bags may have the same card. And there are totally $n$ different cards.

- You want to buy some bags, as few as possible, so that you will have all the different cards.

- (Decision version) Are there $k$ bags whose union has size $n$?
Set Cover

• (Set Cover) Given a set $U$ of $n$ elements (cards), a collection $S_1, \ldots, S_m$ of subsets (bags) of $U$, and a number $k$, is there a collection of at most $k$ of these subsets whose union is equal to $U$?
Set Cover

• (Set Cover) Given a set $U$ of $n$ elements (cards), a collection $S_1, \ldots, S_m$ of subsets (bags) of $U$, and a number $k$, is there a collection of at most $k$ of these subsets whose union is equal to $U$?

• This problem is in NP.

• This problem is NP-hard. We prove this by showing Vertex Cover $\leq_P$ Set Cover.
Set Cover

• (Set Cover) Given a set $U$ of $n$ elements (cards), a collection $S_1, \ldots, S_m$ of subsets (bags) of $U$, and a number $k$, is there a collection of at most $k$ of these subsets whose union is equal to $U$?

• This problem is in NP.

• This problem is NP-hard. We prove this by showing Vertex Cover $\leq_P$ Set Cover.

• Given an instance $(G = (V, E), k)$ of Vertex Cover, construct an instance of Set Cover, as follows.

• Let $U = \{e_1, \ldots, e_m\}$ be the set consisting of the edges in $E$. For each node $v$ in $V$, construct a subset $S_v \subseteq U$ consisting of the edges adjacent to $v$. 
Set Cover

- Let $U = \{e_1, ..., e_m\}$ be the set consisting of the edges in $E$. For each node $v$ in $V$, construct a subset $S_v \subseteq U$ consisting of the edges adjacent to $v$.

$$U = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$$
Set Cover

- Let $U = \{e_1, ..., e_m\}$ be the set consisting of the edges in $E$. For each node $v$ in $V$, construct a subset $S_v \subseteq U$ consisting of the edges adjacent to $v$.

Let $U = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$

$v_1 \{e_1, e_2, e_3\}$
$v_2 \{e_1, e_4\}$
$v_3 \{e_4, e_5\}$
$v_4 \{e_6\}$
$v_5 \{e_6, e_7\}$
$v_6 \{e_3, e_5, e_7\}$
Set Cover

- Let $U = \{e_1, \ldots, e_m\}$ be the set consisting of the edges in $E$. For each nodes $v$ in $V$, construct a subset $S_v \subseteq U$ consisting of the edges adjacent to $v$.

\[ U = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\} \]

- $G$ has a vertex cover of size $k$ iff there exists a collection of $k$ subsets whose union is $U$. 
Set Cover

- Let $U = \{e_1, ..., e_m\}$ be the set consisting of the edges in $E$. For each node $v$ in $V$, construct a subset $S_v \subseteq U$ consisting of the edges adjacent to $v$.

- $G$ has a vertex cover of size $k$ iff there exists a collection of $k$ subsets whose union is $U$. 

Vertex Cover $\{v_1, v_3, v_5\}$ 

$\begin{align*}
v_1 & \{e_1, e_2, e_3\} \\
v_2 & \{e_1, e_4\} \\
v_3 & \{e_4, e_5\} \\
v_4 & \{e_6\} \\
v_5 & \{e_6, e_7\} \\
v_6 & \{e_3, e_5, e_7\}
\end{align*}$
Set Cover

• Let $U = \{e_1, ..., e_m\}$ be the set consisting of the edges in $E$. For each nodes $v$ in $V$, construct a subset $S_v \subseteq U$ consisting of the edges adjacent to $v$.

$$v_1 \{e_1, e_2, e_3\}$$
$$v_2 \{e_1, e_4\}$$
$$v_3 \{e_4, e_5\}$$
$$v_4 \{e_6\}$$
$$v_5 \{e_6, e_7\}$$
$$v_6 \{e_3, e_5, e_7\}$$

Vertex Cover $\{v_1, v_3, v_5\}$

Set Cover $\{e_1, e_2, e_3\} \{e_4, e_5\} \{e_6, e_7\}$

• $G$ has a vertex cover of size $k$ iff there exists a collection of $k$ subsets whose union is $U$. 
Problems

\[ \leq P \]

Circuit Satisfiability Problem.

SAT

3-SAT

Independent Set

Vertex Cover

Set Cover
Hamiltonian Cycle

- (Theorem) Hamiltonian Cycle is NP-complete.

- Hamiltonian Cycle is in NP.
- Hamiltonian Cycle is in NP-hard.
Hamiltonian Cycle

• (Theorem) Hamiltonian Cycle is NP-complete.

• Hamiltonian Cycle is in NP.
• Hamiltonian Cycle is in NP-hard.

• 3-SAT $\leq_P$ Hamiltonian Cycle.

• 8.17, pg 475, *Algorithm Design*, by Kleinberg and Tardos. (pdf available online)

• Vertex Cover $\leq_P$ Hamiltonian Cycle.
• Theorem 34.13. pg 1091, *Introduction to Algorithms* (pdf available online)
Problems

\[ \leq_p \]

- Circuit Satisfiability Problem.
  - SAT
    - 3-SAT
      - Independent Set
      - Hamiltonian Cycle
        - Vertex Cover
          - Set Cover
Problems

\[ \leq_P \]

Circuit Satisfiability Problem.

- SAT
- 3-SAT

\[ \leq_P \]

- Independent Set
- Hamiltonian Cycle
- Vertex Cover
- Hamiltonian Path
- Set Cover
Pseudo-polynomial

- If the input is a **numeric number** $k$, the input size is $O(\log k)$.
- $O((\log k)^c)$ for some constant $c$ is **polynomial**.
- $O(k^c)$ for some constant $c$ is **pseudo-polynomial**.
Numeric Problems

• (Subset Sum) Given a multiset $S$ of $n$ positive integers and an integer $k$, decide if $S$ has a subset in which the sum of the elements is equal to $k$.

• (Multiset) May have duplicate elements.
Numeric Problems

• **(Subset Sum)** Given a multiset $S$ of $n$ positive integers and an integer $k$, decide if $S$ has a subset in which the sum of the elements is equal to $k$.

• (Multiset) May have duplicate elements.

• (Recall) There exists a dynamic programming algorithm solving this problem in $\Omega(kn)$.

• Is this algorithm polynomial? No, it is pseudo-polynomial.
Numeric Problems

- **(Subset Sum)** Given a multiset $S$ of $n$ positive integers and an integer $k$, decide if $S$ has a subset in which the sum of the elements is equal to $k$.

- (Multiset) May have duplicate elements.

- (Recall) There exists a dynamic programming algorithm solving this problem in $\Omega(kn)$.

- Is this algorithm polynomial? No, it is pseudo-polynomial.

- Suppose the input is $S = \{a_1, \ldots, a_n\}$ and $k$.
  - The input size is $\Omega(\sum \log a_i + \log k)$.
  - The algorithm runs in $\Omega(kn)$.
Numeric Problems

• (Subset Sum) Given a multiset $S$ of $n$ positive integers and an integer $k$, decide if $S$ has a subset in which the sum of the elements is equal to $k$.

• (Multiset) May have duplicate elements.

• This problem is in NP. (check this by yourself)
• This problem is NP-hard. We can prove this by showing $3$-SAT $\leq_p$ Subset Sum. (Theorem 34.15)
Numeric Problems

• (Partition) Given a multiset $S$ of $n$ integers, decide if $S$ can be partitioned into two subsets $S_1$ and $S_2$ such that the sum of the numbers in $S_1$ is equal to the sum of the numbers in $S_2$. 
(Partition) Given a multiset $S$ of $n$ integers, decide if $S$ can be partitioned into two subsets $S_1$ and $S_2$ such that the sum of the numbers in $S_1$ is equal to the sum of the numbers in $S_2$.

There exists a dynamic programming algorithm solving this problem in $\Omega(Kn)$ where $K$ is the value of the sum of elements in $S$. (similar to the midterm 1)

Is this algorithm polynomial? No, it is pseudo-polynomial.
Numeric Problems

• (Partition) Given a multiset $S$ of $n$ integers, decide if $S$ can be partitioned into two subsets $S_1$ and $S_2$ such that the sum of the numbers in $S_1$ is equal to the sum of the numbers in $S_2$.

• There exists a dynamic programming algorithm solving this problem in $\Omega(Kn)$ where $K$ is the value of the sum of elements in $S$.

• Is this algorithm polynomial? No, it is pseudo-polynomial.

• Suppose the input is $S = \{a_1, ..., a_n\}$.

• The input size is $\Omega(\sum \log a_i)$. The algorithm runs in $\Omega(n \sum a_i)$.
Numeric Problems

• (Partition) Given a multiset $S$ of $n$ integers, decide if $S$ can be partitioned into two subsets $S_1$ and $S_2$ such that the sum of the numbers in $S_1$ is equal to the sum of the numbers in $S_2$.

• This problem is in NP. (check this by yourself)

• This problem is NP-hard. We can prove this by showing Subset Sum $\leq_P$ Partition.
Numeric Problems

• (Partition) Given a multiset $S$ of $n$ integers, decide if $S$ can be partitioned into two subsets $S_1$ and $S_2$ such that the sum of the numbers in $S_1$ is equal to the sum of the numbers in $S_2$.

• This problem is in NP. (check this by yourself)

• This problem is NP-hard. We can prove this by showing Subset Sum $\leq_P$ Partition.

• For an instance of the subset sum problem, a set $S$ of integers and a number $k$, construct an instance of the partition problem with the input set $S^* = S \cup \{a - 2k\}$ where $a$ is the sum of elements in $S$. 
Numeric Problems

• **(Partition)** Given a multiset $S$ of $n$ integers, decide if $S$ can be partitioned into two subsets $S_1$ and $S_2$ such that the sum of the numbers in $S_1$ is equal to the sum of the numbers in $S_2$.

• This problem is in NP. (check this by yourself)

• This problem is NP-hard. We can prove this by showing Subset Sum $\leq_P$ Partition.

• For an instance of the subset sum problem, a set $S$ of integers and a number $k$, construct an instance of the partition problem with the input set $S^* = S \cup \{a - 2k\}$ where $a$ is the sum of elements in $S$.

• $S = \{1, 2, 3, 4, 6, 7\}$ and $k = 8$. ($a - 2k = 23 - 2 \times 8 = 7$)
Numeric Problems

- (Partition) Given a multiset $S$ of $n$ integers, decide if $S$ can be partitioned into two subsets $S_1$ and $S_2$ such that the sum of the numbers in $S_1$ is equal to the sum of the numbers in $S_2$.

- This problem is in NP. (check this by yourself)
- This problem is NP-hard. We can prove this by showing Subset Sum $\leq_P$ Partition.

- For an instance of the subset sum problem, a set $S$ of integers and a number $k$, construct an instance of the partition problem with the input set $S^* = S \cup \{a - 2k\}$ where $a$ is the sum of elements in $S$.
- $S = \{1, 2, 3, 4, 6, 7\}$ and $k = 8$. ($a - 2k = 23 - 2 \times 8 = 7$)
- $S^* = \{1, 2, 3, 4, 6, 7, 7\}$. Sum$(S^*)=30$. 
Numeric Problems

• *(Partition)* Given a multiset $S$ of $n$ integers, decide if $S$ can be partitioned into two subsets $S_1$ and $S_2$ such that the sum of the numbers in $S_1$ is equal to the sum of the numbers in $S_2$.

• This problem is in NP. (check this by yourself)

• This problem is NP-hard. We can prove this by showing Subset Sum $\leq P$ Partition.

• For an instance of the subset sum problem, a set $S$ of integers and a number $k$, construct an instance of the partition problem with the input set $S^* = S \cup \{a - 2k\}$ where $a$ is the sum of elements in $S$. 

Numeric Problems

• **(Partition)** Given a multiset \( S \) of \( n \) integers, decide if \( S \) can be partitioned into two subsets \( S_1 \) and \( S_2 \) such that the sum of the numbers in \( S_1 \) is equal to the sum of the numbers in \( S_2 \).

• This problem is in NP. (check this by yourself)

• This problem is NP-hard. We can prove this by showing Subset Sum \( \leq \) \( P \) Partition.

• For an instance of the subset sum problem, a set \( S \) of integers and a number \( k \), construct an instance of the partition problem with the input set \( S^* = S \cup \{ a - 2k \} \) where \( a \) is the sum of elements in \( S \).

• The total sum of \( S^* \) is \( 2a - 2k \).

• If \( S \) has a subset \( S_1 \) with sum equal to \( k \), consider \( S_1 \cup \{ a - 2k \} \) and \( S^* \setminus \{ S_1 \cup \{ a - 2k \} \} \)
Numeric Problems

• (Partition) Given a multiset $S$ of $n$ integers, decide if $S$ can be partitioned into two subsets $S_1$ and $S_2$ such that the sum of the numbers in $S_1$ is equal to the sum of the numbers in $S_2$.

• This problem is in NP. (check this by yourself)

• This problem is NP-hard. We can prove this by showing Subset Sum $\leq_P$ Partition.

• For an instance of the subset sum problem, a set $S$ of integers and a number $k$, construct an instance of the partition problem with the input set $S^* = S \cup \{a - 2k\}$ where $a$ is the sum of elements in $S$.

• $S = \{1, 2, 3, 4, 6, 7\}$ and $k = 8$.  \{2, 6\}
Numeric Problems

• **(Partition)** Given a multiset $S$ of $n$ integers, decide if $S$ can be partitioned into two subsets $S_1$ and $S_2$ such that the sum of the numbers in $S_1$ is equal to the sum of the numbers in $S_2$.

• This problem is in NP. (check this by yourself)

• This problem is NP-hard. We can prove this by showing Subset Sum $\leq_P$ Partition.

• For an instance of the subset sum problem, a set $S$ of integers and a number $k$, construct an instance of the partition problem with the input set $S^* = S \cup \{a - 2k\}$ where $a$ is the sum of elements in $S$.

  • $S = \{1, 2, 3, 4, 6, 7\}$ and $k = 8$.  
    $\{2, 6\}$
  • $S^* = \{1, 2, 3, 4, 6, 7, 7\}$. Sum($S^*$)=30.  
    $\{2, 6, 7\} \{1, 3, 4, 7\}$
Numeric Problems

• (Partition) Given a multiset $S$ of $n$ integers, decide if $S$ can be partitioned into two subsets $S_1$ and $S_2$ such that the sum of the numbers in $S_1$ is equal to the sum of the numbers in $S_2$.

• This problem is in NP. (check this by yourself)

• This problem is NP-hard. We can prove this by showing Subset Sum $\leq_P$ Partition.

• For an instance of the subset sum problem, a set $S$ of integers and a number $k$, construct an instance of the partition problem with the input set $S^* = S \cup \{a - 2k\}$ where $a$ is the sum of elements in $S$.

• The total sum of $S^*$ is $2a - 2k$.

• If $S^*$ can be partitioned into two subsets $S_1$ and $S_2$ with the same sum, consider the one containing $\{a - 2k\}$. 
Numeric Problems

• (Partition) Given a multiset $S$ of $n$ integers, decide if $S$ can be partitioned into two subsets $S_1$ and $S_2$ such that the sum of the numbers in $S_1$ is equal to the sum of the numbers in $S_2$.

• This problem is in NP. (check this by yourself)

• This problem is NP-hard. We can prove this by showing Subset Sum $\leq_P$ Partition.

• For an instance of the subset sum problem, a set $S$ of integers and a number $k$, construct an instance of the partition problem with the input set $S^* = S \cup \{a - 2k\}$ where $a$ is the sum of elements in $S$.

• $S = \{1, 2, 3, 4, 6, 7\}$ and $k = 8$.

• $S^* = \{1, 2, 3, 4, 6, 7, 7\}$. $\text{Sum}(S^*)=30$. $\{2, 6, 7\} \{1, 3, 4, 7\}$
Numeric Problems

• (Partition) Given a multiset $S$ of $n$ integers, decide if $S$ can be partitioned into two subsets $S_1$ and $S_2$ such that the sum of the numbers in $S_1$ is equal to the sum of the numbers in $S_2$.

• This problem is in NP. (check this by yourself)

• This problem is NP-hard. We can prove this by showing Subset Sum $\leq_P$ Partition.

• For an instance of the subset sum problem, a set $S$ of integers and a number $k$, construct an instance of the partition problem with the input set $S^* = S \cup \{a - 2k\}$ where $a$ is the sum of elements in $S$.

• $S = \{1, 2, 3, 4, 6, 7\}$ and $k = 8$. \{2, 6\}

• $S^* = \{1, 2, 3, 4, 6, 7, 7\}$. Sum($S^*$)=30. \{2, 6, 7\} \{1, 3, 4, 7\}
Numeric Problems

<table>
<thead>
<tr>
<th>Price</th>
<th>$p_1$</th>
<th>$p_2$</th>
<th>$p_3$</th>
<th>$p_4$</th>
<th>$p_5$</th>
<th>$p_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>$v_1$</td>
<td>$v_2$</td>
<td>$v_3$</td>
<td>$v_4$</td>
<td>$v_5$</td>
<td>$v_6$</td>
</tr>
</tbody>
</table>

The pictures belong to their creators.
### Numeric Problems

#### Price

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$p_1$</td>
<td>$p_2$</td>
<td>$p_3$</td>
<td>$p_4$</td>
<td>$p_5$</td>
</tr>
</tbody>
</table>

#### Value

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$v_1$</td>
<td>$v_2$</td>
<td>$v_3$</td>
<td>$v_4$</td>
<td>$v_5$</td>
</tr>
</tbody>
</table>

- You have totally $B$ dollars.
- You want to buy some cards such that the total price is no larger than $B$ and the total value is as much as possible.
Numeric Problems

• (Knapsack) Given a collection of $n$ pairs of numbers, $(v_1, p_1), \ldots, (v_n, p_n)$ and a value $B$, solve the following problem:

\[
\text{maximize} \quad \sum v_i \cdot x_i \\
\text{subject to} \quad \sum p_i \cdot x_i \leq B \quad \text{and} \quad x_i \in \{0, 1\}
\]
Numeric Problems

• (Knapsack) Given a collection of $n$ pairs of numbers, $(v_1, p_1), \ldots, (v_n, p_n)$ and a value $B$, solve the following problem:

  maximize \[ \sum v_i \cdot x_i \]

  subject to \[ \sum p_i \cdot x_i \leq B \text{ and } x_i \in \{0, 1\} \]

• (Decision version) Considering a threshold $T$, is there a solution with total price no larger than $B$ and total value no less than $T$?

  \[ \sum v_i \cdot x_i \geq T \]
  \[ \sum p_i \cdot x_i \leq B \]
  \[ x_i \in \{0, 1\} \]
Numeric Problems

• (Decision version) Considering a threshold $T$, is there a solution with total price no larger than $B$ and total value no less than $T$?

• This problem is in NP.

• This problem is NP-hard. We prove this by showing Subset Sum $\leq_P$ Knapsack.
Numeric Problems

• (Decision version) Considering a threshold $T$, is there a solution with total price no larger than $B$ and total value no less than $T$?

• This problem is in NP.

• This problem is NP-hard. We prove this by showing Subset Sum $\leq_P$ Knapsack

• Consider an instance of Subset Sum, \{${a_1, ... a_n}$\} and $k$.
• Construct an instant of Knapsack with $v_i=p_i=a_i$ and $B = T = k$.

\[
\sum v_i \cdot x_i \geq T
\]
\[
\sum p_i \cdot x_i \leq B
\]

$x_i \in \{0, 1\}$
Numeric Problems

• (Decision version) Considering a threshold $T$, is there a solution with total price no larger than $B$ and total value no less than $T$?

• This problem is in NP.
• This problem is NP-hard. We prove this by showing Subset Sum $\leq_P$ Knapsack

• Consider an instance of Subset Sum, $\{a_1, \ldots, a_n\}$ and $k$.
• Construct an instant of Knapsack with $v_i=p_i=a_i$ and $B=T=k$.

\[
\begin{align*}
\sum v_i \cdot x_i &\geq T \\
\sum p_i \cdot x_i &\leq B \\
x_i &\in \{0,1\}
\end{align*}
\begin{align*}
\sum a_i \cdot x_i &\geq k \\
\sum a_i \cdot x_i &\leq k \\
x_i &\in \{0,1\}
\end{align*}
\]

It has a solution iff there exists a subset with sum equal to $k$. 

Circuit Satisfiability Problem.

\[ \leq_P \]

SAT

3-SAT

Independent Set

Hamiltonian Cycle

Subset Sum

Vertex Cover

Hamiltonian Path

Partition

Set Cover

Knapsack
Problems

\[ \leq_p \]

- Circuit Satisfiability Problem.
  - SAT
    - 3-SAT
      - Independent Set
        - Vertex Cover
          - Set Cover
      - Hamiltonian Cycle
        - Hamiltonian Path
      - Subset Sum
        - Partition
          - Knapsack

you need to know