- Define: **NP-complete**

- **SUBSET-SUM**

- **VERTEX-EXPANSION**

CPSC 501! Please fill out Canvas "discussion" by tomorrow 11:59 pm (Nov 18). Let me know the topic and preferred date of presentation.
Last time:

If $L$ is in NP, and $L \subseteq \Sigma^*$, $\Sigma = \text{alphabet of } \{ \}$

then we had a way of:

- give $w \in \Sigma^*$, Booker formula, $f(w)$

s.t. ① $w \in L \iff f(w) \in \text{SAT}$

also ② $f(w)$ is of size poly in $w$

③ $f(w)$ can be produced in poly time by a Turing machine.

(e.g., $f$ involved $x_{ijk}$, $i = 0 \ldots N$, $j = 1 \ldots Nt|k|$
$k = 1 \ldots |\Gamma|$, $|N| = \text{poly}(|w|)$)

Defs: If $L, L'$ languages, over alphabets $\Sigma, \Sigma'$, then $L \leq \text{poly } L'$ "$L$ is reducible to $L'$ in polynomial time" if there is a Turing
A function $f: \Sigma^* \to \Sigma'^*$ is poly time computable if there is a Turing machine that on input $w \in \Sigma^*$ writes $f(w)$ on one of its tapes and halts in poly time.

We showed: $L \in \text{NP}$ then $L \leq_p \text{SAT}$ and $L \leq_p \text{3SAT}$.

A function $f: \Sigma^* \to \Sigma'^*$ is poly time computable if there is a Turing machine that on input $w \in \Sigma^*$ writes $f(w)$ on one of its tapes and halts in poly time.
E.g., $\Sigma = \{1, 2, 3\}$, $\Sigma' = \{a\}$

$$f : \{1, 2, 3\}^* \rightarrow \{a\}^*$$

**Def:** We say that $L$ is **NP-complete** if

1. $L \in \text{NP}$, and
2. if $L \in \text{NP}$, then $L \leq_p L$.

E.g., SAT, 3SAT are NP-complete.

Today: **SUBSET-SUM** is NP-complete, so is **VERTEX-EXPANSION**, **VERTEX COVER**, **... 3 COLOUR...**

So $\text{NP} = \text{P}$ iff 3SAT $\in \text{P}$.

Also, if $L$ is NP-complete, then $\text{NP} = \text{P} \iff L \in \text{P}$.

Now: to show **BLAH** is NP-complete:

enough to show 1) **BLAH** $\in \text{NP}$, 2) 3SAT $\leq_p \text{BLAH}$.

or any other NP-complete language
SUBSET-SUM (an "arithmetic" problem)

\[ \{ \{ x_1, \ldots, x_m, t \} \mid \text{m} \in \mathbb{N}, \text{ } x_1, \ldots, x_m, t \in \mathbb{N} \text{ and for some } \]
\[ I \subseteq \{ m \} , \]
\[ \sum_{i \in I} x_i = t \]

Idea: you have jobs that take time \( x_1, \ldots, x_m \) and a computer/bin/… that has time \( t \) to run.

Related problems in "load balancing" "bin packing"…

Claim! Take \( f \in 3\text{CNF}, f = f(x_1, \ldots, x_k) \) and is written as

\[ C_1 \text{ AND } C_2 \text{ AND } \ldots \text{ AND } C_k, \text{ each } C_i \text{ OR of 3 literals} \]

Take \( F = (x_1 \text{ OR } x_2 \text{ OR } \overline{7}x_3) \text{ AND } (x_2 \text{ OR } \overline{7}x_1 \text{ OR } x_3) \)
reduce $F$ to a subset sum problem.

Idea! $F$ is satisfiable iff for some $x_1 = T$, $F$

```
  \{ \text{gadget} \}
  \begin{align*}
  x_2 = T, F \\
  x_3 = T, F
  \end{align*}
```

base 10
a digit
number

Each $C_i$ is $T$

<table>
<thead>
<tr>
<th>$y_1$</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_2$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$y_3$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$y_4$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$y_5$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Dummy variables for clause 1

$|v_1| = 1$, $|v_2| = 2$

Target: $3(4)$
So one of $y_1, z_1$ will be chosen

\[- \quad y_2, z_2 \quad \vDash \quad \]

clause_1 = $x_1 \lor x_2 \lor \neg x_3$

take $y_1, y_2, z_3$ (at least one)

break 10:28-10:33
\((X_1 \text{ or } X_2 \text{ or } \neg X_3) \text{ AND } (\neg X_2 \text{ or } \neg X_1 \text{ or } X_3)\)

\[
\begin{array}{cccc}
X_1 & X_2 & X_3 & C_1 & C_2 \\
\hline
y_1 & 1 & 0 & 0 & 1 & 0 \\
n_1 & 1 & 0 & 0 & 0 & 1 \\
y_2 & 0 & 1 & 0 & 1 & 1 \\
n_2 & 0 & 1 & 0 & 0 & 0 \\
y_3 & 0 & 0 & 1 & 0 & 0 \\
n_3 & 0 & 0 & 1 & 1 & 0 \\
y_4 & 0 & 0 & 0 & 1 & 0 \\
n_4 & 0 & 0 & 0 & 0 & 1 \\
\hline
t & 1 & 1 & 1 & 3 & 3 \\
\end{array}
\]

1. Build something in \textit{SUBSET-SUM} represents \(X_1 = T\) or \(X_1 = F\), \(X_2 = T\) or \(F\), \ldots, \(X_3 = T\) or \(F\).

2. Build something in \textit{SUBSET-SUM} to check if \(C_1, C_2, \ldots, C_k\) are all satisfied.

\[
\{ \emptyset, 0010, 10001, 01011, 01000, 01133, 00001, 00011, 11133 \} \]
\[ x_1 - - x_k \leq c_1 - - c_n \]

\[ y_1, z_1, \ldots, y_k, z_k \]

\[ \# \text{digits} \geq k \]

\[ \# \text{of numbers} \]

\[ 2^{l+2k+1} \]

Site problem: \((l+k)(2^{l+2k+1})\)

3CNF has \(l\) vars, \(k\) clauses
Next time!

VERTEX - EXPANSION is NP-complete —

Class ends
\[ L = \left\{ \langle m \rangle \mid \exists \text{ one of its inputs such that } \langle m \rangle \text{ accepts at least } \right\} \]

Imagine you have a \( U = \text{ universal TM} \).

\( \langle m \rangle \) runs over \( \Sigma = \{1, 2, \ldots, 1 \leq 1\} \).

Say \( \Sigma = \{1, 2\} \).

Possible inputs: \( E_3, 1, 2, 11, 12, 21, 22, \ldots \).

Step 1

Step 2

back