Informal Summary from Last Time

1. Decision problem $\Pi$ is in $P$ if there is a polynomial time algorithm that correctly answers $\Pi$

2. Decision problem $\Pi$ is in $NP$ if there is a polynomial time algorithm that takes advice:
   - If the answer should be “yes”, then there exists advice that leads the algorithm to output “yes”
   - If the answer is “no”, then there doesn’t exist advice that would lead the algorithm to output “yes”

3. A problem $\Pi$ is NP-hard if for any $\Pi' \in NP$: $\Pi' \leq_P \Pi$

4. A problem $\Pi$ is NP-complete if $\Pi \in NP$ and $\Pi$ is NP-hard

5. To show $\Pi$ is NP-complete it suffices to show that
   - $\Pi$ is in NP
   - $\Pi' \leq_P \Pi$ for some $\Pi'$ that is already known to be NP-hard

6. It’s widely believed the $P \neq NP$ but finding a polynomial time algorithm for any NP-hard problem would prove $P = NP$.  

2/6
Problem: Subset-Sum

- **Input:** A set $S$ of $n$ integers $\{s_{1}, s_{2}, \ldots, s_{m}\}$ and a target integer $t$.
- **Question:** Is there a subset $S' \subset S$ such that $t = \sum_{s \in S'} s$?
Subset-Sum is NP-Complete

Theorem

Subset-Sum is NP-Complete

Proof.

1. Easy to show Subset-Sum is in NP
2. It suffices to show 3-SAT ≤\text{p} Subset-Sum
3. Given
   \((l_{1,1} \lor l_{1,2} \lor l_{1,3}) \land (l_{2,1} \lor l_{2,2} \lor l_{2,3}) \land \ldots \land (l_{m,1} \lor l_{m,2} \lor l_{m,3})\) in
   \(n\) variables, define the set of integers (expressed in decimal):

   \[
   \text{For } i \in [n]: s_i = (1, 0, \ldots, 0, y_m, \ldots, y_1), \quad s'_i = (1, 0, \ldots, 0, z_m, \ldots, z_1)
   \]

   where \(y_j = 1\) if \(x_i\) is a literal in \(j\)-th clause and 0 otherwise
   and \(z_j = 1\) if \(\overline{x_i}\) is a literal in \(j\)-th clause and 0 otherwise.

   \[
   t = (1, \ldots, 1, 3, \ldots, 3)
   \]
Suppose \( \phi \) is satisfiable:

1. Fix a satisfying assignment
2. Let \( S' = \{s_i : x_i = \text{TRUE}\} \cup \{s'_i : x_i = \text{FALSE}\} \)
3. So far, for some \( a_i \geq 1 \):
   \[
   \sum_{s \in S'} s = (1, \ldots, 1, a_m, a_{m-1}, \ldots, a_1)
   \]
4. Can add “\( h \)” elements to \( S' \) such that
   \[
   \sum_{s \in S'} s = (1, \ldots, 1, 3, \ldots, 3) = t
   \]
Subset $S'$ that sums to $t$ implies $\phi$ is satisfiable

Suppose $\sum_{s \in S'} s = t$:

1. For each $i \in [n]$, exactly one of $s_i$ and $s'_i$ are in $S'$
2. Let $x_i$ be TRUE if $s_i \in S'$ and FALSE otherwise
3. Since there are only two "h" elements corresponding to each clause, each clause must be satisfied.