Subset-Sum:

- **input:** Positive integers $a_1, \ldots, a_n$ & $t$
- **output:** subset $S$ of objects $\{1, \ldots, n\}$ where $\sum_{i \in S} a_i = t$
  - if such a $S$ exists
  - **NO** otherwise

Can solve in $O(nt)$ time.

But Subset-Sum is NP-complete

**Proof of:**

a) Subset-Sum is NP:
   - given inputs $a_1, \ldots, a_n$, $t$ & $S$
   - in $O(n)$ time can check that $\sum_{i \in S} a_i = t$

b) 3SAT $\rightarrow$ Subset-Sum
   - Take input $f$ for 3SAT
   - Variables $x_1, \ldots, x_n$
   - Clauses $C_1, \ldots, C_m$
Basic assumptions about f:
- No clause contains $x_i \& \overline{x_i}$; otherwise it's satisfied & we can drop it.
- Each $x_i$ is in at least 1 clause; otherwise set $x_i = F$ & reduce similarly each $\overline{x_i}$ is in $\geq 1$ clause.

The input to subset-sum will be numbers:

$v_1, v_2, \ldots, v_n, v_n, s_1, s_1, \ldots, s_m, s_m$

& $t$ will be an $n+m$ digit number

All numbers will be base 10.

$v_i$ corresponds to $x_i$; $v_i \in S$ then $x_i = T$
$v_i \in \overline{S}$ to $x_i$; $v_i \in S$ then $x_i = F$

We need that exactly 1 of $v_i, v_i \in S$, is in $S$.
To achieve this: in the $i^{th}$ digit of $v_i, v_i \in S$, put a 1
All other numbers have a 0 in $i^{th}$ digit.
Digit $n+j$ corresponds to clause $C_j$.

If $x_i \in C_j$, put a 1 in digit $n+j$ for $x_i$.
If $\overline{x}_i \in C_j$, put a 1 in digit $n+j$ for $x_i$.

Want that 1, 2 or 3 of literals in $C_j$ are included in $S$.

$\Rightarrow$ Put a 3 in digit $n+j$ of $+$.

Use $S_j, \overline{S}_j$ as buffers.

$\Rightarrow$ Put a 1 in digit $n+j$ of $S_j, \overline{S}_j$.

$\Rightarrow$ Put a 0 in digit $n+j$ of all other numbers not yet defined here.

To get a sum of 3 in digit $n+j$ need to include

1 literal of $C_j$ + 1 of $S_j, \overline{S}_j$

or 3

0
Example:

\[ f = (x_1 \lor \overline{x_2} \lor \overline{x_3}) \land (\overline{x_1} \lor \overline{x_2} \lor x_3) \land (x_1 \lor \overline{x_2} \lor x_3) \land (x_1 \land x_2) \]

<table>
<thead>
<tr>
<th></th>
<th>(x_1)</th>
<th>(x_2)</th>
<th>(x_3)</th>
<th>(C_1)</th>
<th>(C_2)</th>
<th>(C_3)</th>
<th>(C_4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(V_1)</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>(V_1')</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(V_2)</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>(V_2')</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>(V_3)</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>(V_3')</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(S_1)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(S_1')</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>(S_2)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(S_2')</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(S_3)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>(S_3')</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>(S_4)</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>(S_4')</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>+</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Total count of 1's and 0's in the table.
Subset-Sum has a solution iff 3SAT f is satisfiable.

\( \Rightarrow \) For 1st \( n \) digits, include \( \overline{v}_i \) or \( v_i \) (but not both) to get a 1 in digit \( i \):

- if \( v_i \in S \Rightarrow x_i = T \)
- if \( \overline{v}_i \notin S \Rightarrow x_i = F \)

For digit \( n+j \):

- to get a sum of 3 need to include \( \geq 1 \) of literals in \( C_j \)

So \( C_j \) is satisfied.

\( \Leftarrow \) if \( x_i = T \), add \( v_i \) to \( S \)
- if \( x_i = F \), add \( \overline{v}_i \) to \( S \)

\( \Rightarrow \) so i-th digit of \( \overline{f} \) is satisfied.

For clause \( C_j \) at least 1 literal is satisfied then add \( S_j \) &/or \( \overline{S}_j \) if needed to get a sum of 3 in digit \( n+j \).
Knapsack:

- Input: \( n \) objects with weights \( w_1, \ldots, w_n \) & values \( v_1, \ldots, v_n \)
- Capacity \( B \)
- Value \( V \)

Output: subset \( S \) of objects with

\[
\sum_{i \in S} w_i \leq B
\]

&

\[
\sum_{i \in S} v_i \geq V
\]

& NO if no such \( S \) exists

Knapsack is NP-complete

Proof:

a) Knapsack is in NP:

Given input to knapsack \( S \)
then in \( O(n) \) time can check
that \[
\sum_{i \in S} w_i \leq B \& \sum_{i \in S} v_i \geq V.
\]
b) SubsetSum $\Rightarrow$ Knapsack

Take input $a_1, \ldots, a_n$ & $t$ for subset-sum

Set $V_i = W_i = a_i$

Set $B = V = t$

Then Run knapsack on $w_1, \ldots, w_n$

$V_i, \ldots, V_n$

$B, V$

We're trying for a subset $S$ where:

\[ \sum_{i \in S} w_i \leq B \iff \sum_{i \in S} a_i \leq t \]

\[ \sum_{i \in S} V_i \geq V \iff \sum_{i \in S} a_i \geq t \]

\[ \sum_{i \in S} a_i = t \leq \sum_{i \in S} a_i \]

So same as subset-sum.