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 $\underbrace{\text{Subset-sum is NP-complete}}$

Proof of $\nearrow$:

a) Subset-Sum $\in$ 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 ≥ 1 clause

The input to subset-sum will be numbers:
$v_1, v_1', \ldots, v_n, v_n', s_1, s_1', \ldots, s_m, s_m'$
& t will be a $n+m$ digit number
all numbers will be base 10.

v_i corresponds to x_i ; $v_i \in S$ then $x_i = T$
v_i' " to $\overline{x_i}$; $v_i' \in S$ then $x_i = F$
— need that exactly 1 of v_i, v_i' is in S.
To achieve this: in the i^{th} digit
of v_i, v_i' & t 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 v_i

if $\overline{x}_i \in C_j$, put a 1 in digit $n+j$ for v_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 t

Use s_j, s_j' as buffers:

$\Rightarrow$ Put a 1 in digit $n+j$ of s_j & 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		2
2	literals of C_j +	1 of s_j, s_j'
or 3		0

Example:

$$f=(\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3) \wedge (\overline{x}_1 \vee \overline{x}_2 \vee x_3) \wedge (x_1 \vee \overline{x}_2 \vee x_3) \wedge (x_1 \vee x_2)$$

	x_1	x_2	x_3	C_1	C_2	C_3	C_4	
v_1	1	0	0	0	0	1	1	$v_1=1000011$
v_1'	1	0	0	1	1	0	0	$v_1'=1001100$
v_2	0	1	0	0	0	0	1	$v_2=100001$
v_2'	0	1	0	1	1	1	0	$v_2'=101110$
v_3	0	0	1	0	1	1	0	$v_3=10110$
v_3'	0	0	1	1	0	0	0	$v_3'=11000$
S_1	0	0	0	1	0	0	0	$S_1=1000$
S_1'	0	0	0	1	0	0	0	$S_1'=1000$
S_2	0	0	0	0	1	0	0	$S_2=100$
S_2'	0	0	0	0	1	0	0	$S_2'=100$
S_3	0	0	0	0	0	1	0	$S_3=10$
S_3'	0	0	0	0	0	1	0	$S_3'=10$
S_4	0	0	0	0	0	0	1	$S_4=1$
S_4'	0	0	0	0	0	0	1	$S_4'=1$
t	1	1	1	3	3	3	3	$t=1113333$

Subset-Sum has a solution iff 3SAT f is satisfiable

($\Rightarrow$) For 1^{st} n digits,
include 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 $v_i' \in S \Rightarrow x_i = F$

For digit $n+j$,
to get a sum of 3 need to
include ≥ 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 v_i' to S
$\Rightarrow$ so i^{th} digit of t is satisfied
For clause C_j at least 1 literal is satisfied
then add s_j &/or s_j' if needed
to get a sum of 3 in digit $n+j$

Knapsack:

input: n objects with weights $w_1, \dots, w_n$
& values $v_1, \dots, v_n$

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 $\in$ 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) Subset Sum $\rightarrow$ 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 $w_1, \ldots, w_n$
$v_1, \ldots, v_n$,
B, V

We're trying for a subset S where:

$$\sum_{i \in S} w_i \le B \iff \sum_{i \in S} a_i \le t$$

$$\sum_{i \in S} v_i \ge V \iff \sum_{i \in S} a_i \ge t$$

$$\sum_{i \in S} a_i = t$$

So same as subset-sum.