Today: Knapsack (search version) is NP-complete.

Main task: Subset-sum is NP-complete.

Subset-sum:

input: Positive integers $a_1, \ldots, a_n$ & t

output: subset S of objects $\{1, \ldots, n\}$

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

& NO if no such S exists.

Using DP, can solve in $O(nt)$ time.

Proof that Subset-sum is NP-complete:

a) Subset-sum $\in$ NP:

Given inputs $a_1, \ldots, a_n, t$ & S, in $O(n)$ time we can check that $\sum_{i \in S} a_i = t$

b) 3SAT → Subset-sum:

Take input f for 3SAT
where f has variables $x_1, \ldots, x_n$
& clauses $C_1, \ldots, C_m$

Let's make some 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 ≥ 1 clause, otherwise set $x_i = F$ & simplify

similarly, each $\overline{x_i}$ is in ≥ 1 clause.

We'll define an instance of subset-sum which has the following input:

$2n+2m$ numbers: $v_1, v_1', \ldots, v_n, v_n', s_1, s_1', \ldots, s_m, s_m'$
& t

all numbers will be base 10 & $\leq n+m$ digits
(note: the numbers are $\leq 10^{n+m}$
So they are HUGE)

Intuition: v_i corresponds to x_i: $v_i \in S \Leftrightarrow x_i = T$

v_i' " $\overline{x_i}$: $v_i' \in S \Leftrightarrow x_i = F$

— Hence we want that exactly 1 of v_i, v_i' are in S.

To achieve this:

in digit i:

put a 1 for v_i, v_i' & t

& put a 0 for all other numbers.

(assuming no carries between digits, then this ensures exactly one of v_i, v_i' are in a solution S)

For clause C_j use digit $n+j$:

if $x_i \in C_j$ put a 1 in v_i

if $\overline{x_i} \in C_j$ put a 1 in v_i'

Want that 1, 2, or 3 of these literals are included in S

So put a 3 in t

Use s_j, s_j' as buffers (in case only 1 or 2 literals included)

put a 1 in s_j, s_j'

all other numbers have 0 in digit $n+j$.

Note: To get a sum of 3 in Digit $n+j$ need to include in S:

a) $=1$ literal of C_j & $s_j + s_j'$

b) $=2$ literals of C_j & s_j or s_j' (but not both)

c) $=3$ literals of C_j & none of s_j, s_j'

---

all are ≥ 1 literal of C_j so C_j satisfied.

Let's look at an example.

Example: $f = \overset{C_1}{(\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})} \wedge \overset{C_2}{(\overline{x_1} \vee \overline{x_2} \vee x_3)} \wedge \overset{C_3}{(x_1 \vee \overline{x_2} \vee x_3)} \wedge \overset{C_4}{(x_1 \vee x_2)}$

digit 1 corresponds to x_1

	x_1	x_2	x_3	C_1	C_2	C_3	C_4	
v_1	1					1	1	$v_1 = 1000011$
v_1'	1			1	1			$v_1' = 1001100$
v_2		1					1	$v_2 = 0100001$
v_2'		1		1	1	1		$v_2' = 0101110$
v_3			1		1	1		$v_3 = 0010110$
v_3'			1	1				$v_3' = 0011000$
S_1				1				$S_1 = 0001000$
S_1'				1				$S_1' = 0001000$
S_2					1			$S_2 = 0000100$
S_2'					1			$S_2' = 0000100$
S_3						1		$S_3 = 0000010$
S_3'						1		$S_3' = 0000010$
S_4							1	$S_4 = 0000001$
S_4'							1	$S_4' = 0000001$
t	1	1	1	3	3	3	3	$t = 1113333$

<u>Lemma</u>: Subset-sum has a solution S iff 3SAT f has a satisfying assignment

<u>Proof</u>:

($\Rightarrow$) For 1^{st} n digits,

S must include v_i or v_i' (but not both)

to get a 1 in digit i (as required by t)

if $v_i \in S \Rightarrow$ set $x_i = T$

if $v_i' \in S \Rightarrow$ set $x_i = F$

For digit $n+j$

to get a sum of 3 need to include

≥ 1 literal from C_j

so C_j is satisfied by the above assignment.

($\Leftarrow$) if $x_i = T$ add v_i to S & if $x_i = F$ add v_i' to S.

Hence i^{th} digit of T is satisfied.

For clause C_j ≥ 1 literal is satisfied

and add s_j &/or s_j' to get up to a sum of 3.

∎

This proves that the subset-sum instance we created has a solution iff the original 3SAT input f has a satisfying assignment.
And given a solution S to subset-sum we saw how to get the satisfying assignment.
This finishes the NP-completeness proof for Subset-sum.

Knapsack: (search version)

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

capacity B
value V

output: subset S of objects with

$$\sum_{i \in S} w_i \le B$$

$$\& \sum_{i \in S} v_i \ge V$$

& NO if no such S exists.

Knapsack is NP-complete

a) Knapsack $\in NP$

Consider input $\{w_1, \dots, w_n, v_1, \dots, v_n, B, V\}$ & solution S.
Let's assume the input size is N.
So $n \leq N$ & all numbers are $\leq 2^N$.
Then in $O(N^2)$ time we 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 $a_1, \dots, a_n$ & t for subset-sum.

Set $v_i = w_i = a_i$.

Set $B = V = t$.

Run Knapsack on $\{w_1, \dots, w_n, v_1, \dots, v_n, B, V\}$
A solution S has

$$\left.\begin{aligned} \sum_{i \in S} w_i \leq B &\Rightarrow \sum_{i \in S} a_i \leq t \\ \& \sum_{i \in S} v_i \geq V &\Rightarrow \sum_{i \in S} a_i \geq t \end{aligned}\right\} \sum_{i \in S} a_i = t$$

So it's a solution for Subset-sum. ∎