Knapsack Problem (0-1)

We have n items $t_1, t_2, \dots, t_n$ and each item has a weight w_i and a value. We would like to select a subset of items such that the total weight is $\leq W$ ($\leftarrow$ a given capacity) a total value is maximal.

We select $I \subseteq \{1, 2, \dots, n\}$ such that

$$\sum_{i \in I} w_i \leq W$$

$$\sum_{i \in I} v_i \longrightarrow \max$$

$S[i,j]$: subproblem of choosing items only from $\{t_1, t_2, \ldots, t_i\}$ and having capacity j

no items

$0 \le i \le n$

$0 \le j \le W$

$c[i,j] :=$ the maximal total value in $S[i,j]$

Because of the <u>optimal substructure property</u> (see lecture) we have the following recursive formula

$$c[i,j] = \begin{cases} 0 & \text{if } i=0 \text{ or } j=0 \\ c[i-1,j] & \text{if } i \geq 1 \text{ and } j \geq 1 \\ & \quad \text{and } w_i > j \\ \max\{\underbrace{c[i-1,j]}_{\substack{t_i \text{ not in the} \\ \text{optimal solution}}}, \underbrace{c[i-1,j-w_i]+v_i}_{\substack{t_i \text{ is included} \\ \text{in the optimal solution}}}\} & \text{else} \end{cases}$$

Example

$W = 10$

	1	2	3	4	5	6	7
w_i	4	5	3	1	1	4	3
v_i	40	60	10	10	3	20	60

$(n+1) \times (W+1)$

* if t_i is chosen

v_i	w_i		$j=0$	1	2	3	4	5	6	7	8	9	10
		$i=0$	0	0	0	0	0	0	0	0	0	0	0
40	4	1	0	0	0	0	40	40*	40*	40*	40*	40*	40*
60	5	2	0	0	0	0	40	60*	60*	60*	60*	100*	100*
10	3	3	0	0	0	10*	40	60	60	60	70*	100	100
10	1	4	0	10*	10*	10(*)	40	60	70*	70*	70(*)	100	110*
3	1	5	0	10	13*	13*	40	60	70	73*	73*	100	110
20	4	6	0	10	13	13	40	60	70	73	73	100	110
60	3	7	0	10	13	60*	70*	73*	73*	100*	120*	130*	133*

max value 133

$\{t_2, t_4, t_5, t_7\}$

$5 + 1 + 1 + 3$

$60 + 10 + 3 + 60$

Greedy algorithm would be: choose item with maximal v_i/w_i, if there is room for more items, then again choose the item with max v_i/w_i ... so on

In previous example:

	t_1	t_2	t_3	t_4	t_5	t_6	t_7
w_i	4	5	3	1	1	4	3
v_i	10	60	10	10	3	20	60
v_i/w_i	2.5	12	3.3	10	3	5	20

t_7, t_2, t_4 ~~t_6~~ ~~t_3~~ t_5

$3 + 5 + 1 + 1 = 10$

Same solution as before

But: the greedy algorithm is **not** always good.

Example when it is **bad**:

	t_1	t_2	t_3
w_i	4	2	3
v_i	100	41	61
v_i/w_i	25	20.5	20.5

$W = 5$

It gives $\{t_1\}$ but $\{t_2, t_3\}$ would be better

$c[i,j]$: what is the maximal value $\sum_{i \in I} v_i$

with $I \subseteq \{1, 2, \ldots, i\}$ and $\sum_{i \in I} w_i \leq j$

If I can choose from $\{t_1, t_2, \ldots, t_i\}$ such that the total weight $\leq j$ then the max total value I can achieve is $c[i,j]$

Longest common subsequence = $\boxed{LCS}$

$X = x_1 \, x_2 \, x_3 \, \dots \, x_n$ $\qquad I \subseteq \{1, 2, \dots, n\}$

$Y = y_1 \, y_2 \, y_3 \, y_4 \, y_5 \, \dots \, y_{m-1} \, y_m$

common subsequence

We are looking for the longest one.

$X = abbcabccab$

$Y = caabccbacb$

$S[i,j] =$ finding LCS of $X_i = x_1 x_2 x_3 \dots x_i$

and $Y_j = y_1 y_2 y_3 \dots y_j$

$i=0 \rightarrow$ empty string

$j=0 \rightarrow$ —— " ——

$d[i,j] :=$ length of the optimal solution of $S[i,j]$

Because of the optimal substructure property

$$d[i,j] = \begin{cases} d[i-1,j-1]+1 & \text{if } x_i = y_j \\ \max\{d[i-1,j], d[i,j-1]\} & \text{if } x_i \neq y_j \end{cases}$$

$d[i,0] = 0$

$d[0,j] = 0$

$X = abbcabccab$

$Y = caabccbacb$

	0	c	a	a	b	c	c	b	a	c	b
0	0	0	0	0	0	0	0	0	0	0	0
a	0	0	1	1	1	1	1	1	1	1	1
b	0	0	1	1	2	2	2				
b	0										
c	0										
a											
b											

→ When diagonal step "↖", then that character is in the opt. subsequence.

length of opt. solution here

What is the subsequence?

Backtrack along arrows to find the longest subsequence