KNAPSACK AUCTION

KNAPSACK PROBLEM

INPUT $I = \{1, \dots, n\}$ items

s_i = size for each item, $\in N^+$

w_i = value for each item $\in N^+$

C = maximum capacity of the knapsack, $\in N^+$

QUESTION which items should be selected to maximise the value but not exceeding C?

Formally: $\arg\max_{I' \subseteq I} \sum_{i \in I'} w_i$

s.t. $\sum_{i \in I'} s_i \leq C$

Algorithm

	1	2	3	4	5	6		C
i_1			w_1					
i_2		w_2	w_1		w_1, w_2			
i_3	w_3	w_2	w_1		w_1, w_2	w_1, w_2, w_3		

$w_1 = 3$

$w_2 = 2$

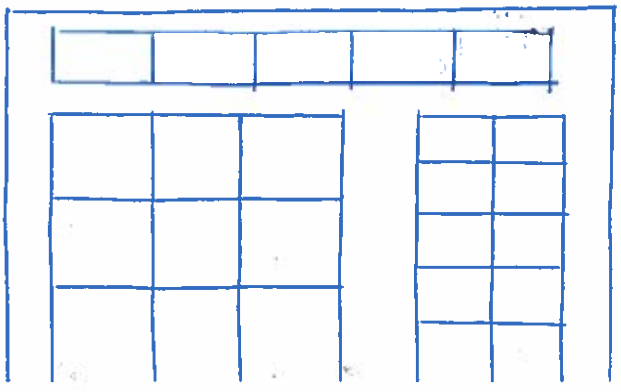
$\dots$ $w_3 = 1$

It is a dynamic programming algorithm: given a row, we copy it in the next one and add the item to each allocation. If there are no allocation with the same weight we keep the most valuable. At the end, we extract from the last row the allocation with the highest value.

For every item, we produce at most C elements: $O(n\,C)$: where is the hardness? to represent C we need n bits, it's a natural number, so $C = 2^n$: it is exponential!

✱ (vedi dietro)

We can adopt this algorithm for auctions $w_i = \theta_i$, the value of the ad



s_i	5	2	7	1	3
w_i	7	4	10	3	5
	i_1	i_2	i_3	i_4	i_5

$C = 12$

	1	2	3	4	5	6	7	8	9	10	11	12
i_1					w_1							
i_2		w_2			w_2		w_1+w_2					
i_3		w_2			w_1		w_1+w_2		w_2+w_3			w_1+w_3
i_4	w_4	w_2	w_2+w_4		w_1	w_1+w_4	w_1+w_2	$w_1+w_2+w_4$	w_2+w_3	$w_2+w_3+w_4$		w_1+w_3
i_5	w_4	w_2	w_2+w_4		w_1	w_1+w_4	w_1+w_2		$w_2+w_4+w_5$	$w_2+w_3+w_4$		
tot	3	4	7		7	10	11	14	15	17		17

$w_2+w_4 > w_5$

$w_1+w_2 > w_3$

$w_2+w_3 < w_1+w_4+w_5$

$w_2+w_3+w_4 > w_1+w_2+w_5$

$p_1 = 17-(17-7) = 17-10 = 7$

$p_3 = 17-(17-10) = 17-7 = 10$

$p_2 = p_4 = p_5 = 0$

Best allocations: i_2, i_3, i_4 ; i_1+i_3 ——————→

⊛ KNAPSACK AUCTION: a knapsack auction is defined as:

- $N = \{1, \dots, n\}$ players;
- $I = \{1, \dots, n\}$ items;
- $s = \{s_1, \dots, s_n\}$, $s_i \in \mathbb{N}^+$, size of items
- $w = \{w_1, \dots, w_n\}$, $w_i \in \mathbb{N}^+$, value of item i and private info of player i.
- C = capacity
- $\Theta_i \subset \mathbb{N}^+$, space of types of player i; $\theta_i = w_i$
- $K \subseteq I$ is the set of feasible allocations s.t. $\sum_{i \in K} s_i \leq C$
- $v_i(K, \theta_i) = \begin{cases} \theta_i, & i \in K \\ 0, & \text{otherwise} \end{cases}$

- VCG mechanism can be applied to the knapsack auction, requiring an algorithm to find the exact solution of knapsack, but no polynomial time algorithm exists for such problem unless $P = NP$