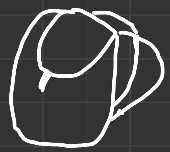
Dynamic Programming : Knapsack



n-items

weight: $a_1, \dots, a_n \in \mathbb{N}$

profit: $p_1, \dots, p_n \in \mathbb{N}$

Capacity $D \in \mathbb{N}$

Task: Choose $S \subseteq \{1, \dots, n\}$ s.th.

$$\sum_{i \in S} a_i \leq D$$

and

$$\sum_{i \in S} p_i \quad \text{MAXIMAL.}$$

$$\max \sum_{i=1}^{n} p_i x_i$$

$$\text{s.th.} \quad \sum_{i=1}^{n} a_i x_i \leq D$$

$$x_i \in \{0, 1\} \quad i = 1, \dots, n$$

NP HARD BUT There exists "pseudopolynomial" time algorithm.

1.) $P(n, D)$

2.) $P(n, P)$

$P = \max \{p_i\}$

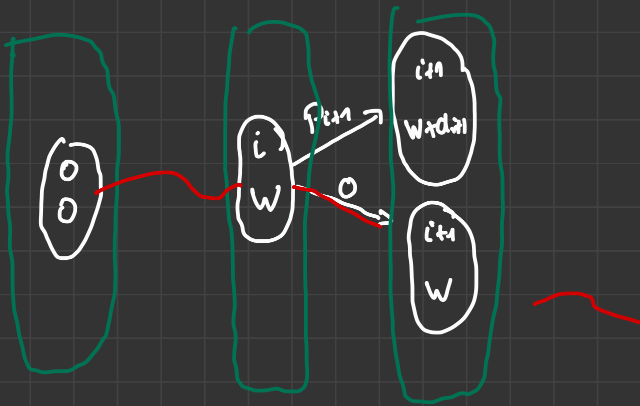
If weights $3 : |||$ $4\ ||||$

Profits $7 : |||||||$

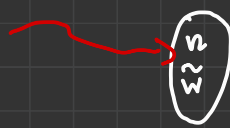
Dynamic program I

$\bar{x} \in \{0,1\}^n$ optimal solution.

consider $a_1 \bar{x}_1 + \dots + a_i \bar{x}_i$: total weight accumulated by first i items.



Build this directed acyclic graph



$\tilde{W} \le D$

$\bar{x} \approx$ opt. sol. found by longest path.

$\bar{x}_{i+1} = 0$

#Vertices of this graph: $n \cdot D$

ARCS: $\le 2 \cdot$ #VERTICES: $O(n \cdot D)$

$\Rightarrow$ Longest Path can be computed in time $O(n \cdot D)$.

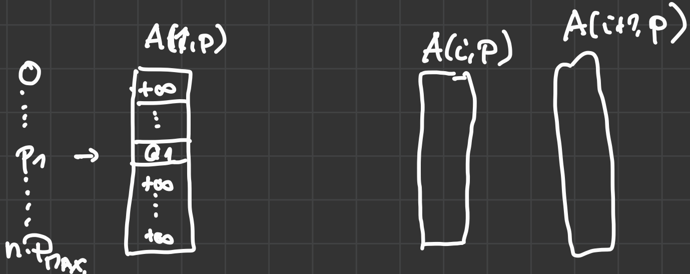
$\boxed{\text{Thm: } \Rightarrow \text{ Knapsack Problem can be solved in time } O(n \cdot D)}$

Dynamic Program II

$A(i, p)$ Smallest weight subset $\{1, \dots, i\}$

that have total profit p.

$$A(1, p) = \begin{cases} a_1 & \text{if } p = p_1 \\ +\infty & \text{otherwise.} \end{cases}$$



$P_{MAX} = \max_i \{p_i\}$

$$A(i+1, p) = \begin{cases} \min \{ A(i,p),\ a_{i+1} + A(i, p - p_{i+1}) \} & \text{if } p_{i+1} < p \\ A(i,p) & \text{otherwise} \end{cases}$$

Record the min-weight sets!

<u>Thm:</u> Let $P_{MAX} = \max \{p_i\}$

The Knapsack problem can be solved in time

$$O(n \cdot n \cdot P_{MAX}) = O(n^2 \cdot P_{MAX}).$$

<u>Pseudopolynomial!!</u>

A FPTAS for Knapsack:

Given $\varepsilon > 0$ TASK $p(S) \geq (1-\varepsilon) \cdot OPT.$

Algorithm: $p(\frac{1}{\varepsilon}, n)$

$$K := \frac{\varepsilon \cdot p_{max}}{n}$$

New profits: $p' := \left\lfloor \frac{p}{K} \right\rfloor$

$$\|p'\|_\infty \leq \frac{n}{\varepsilon}.$$

$\Rightarrow$ 2.nd Algorithm, if run on p' has running time

$$\frac{n^3}{\varepsilon}$$

Approximation Alg: Run 2.nd Dyn. Prog on p'.

Return opt solution $S \subseteq \{1, \dots, n\}$ for p'.

Thm: $p(S) \geq (1-\varepsilon) \cdot OPT$

Proof: $O \subseteq \{1, \dots, n\}$. $|O| \leq n$

$$p(O) - K \cdot p'(O) \leq p(O) - p(O) + n \cdot K$$

$$= n \cdot K$$

$$p(S) \geq K \cdot p'(S) \geq K \cdot p'(O) \geq p(O) - n \cdot K$$

$p_{MAX} \leq OPT$

$$= OPT - \varepsilon \cdot p_{MAX} \geq OPT(1-\varepsilon)$$