○ Ripasso problema di knapsack → formulazione ricorsiva più precisa sull'e-learning (RIPASSO).

voglio calcolare $OPT_{i,c}$ $\forall (i,c)$

PD_OPT_KNAPSACK $(n, C, v_1, v_2 \dots v_n, w_1, w_2 \dots, w_n)$:

```
for i=0 to n :          caso b. 1
    OPT_{i,0} = 0 ;
for c=0 to C :          caso b. 2
    OPT_{0,c} = 0 ;
for i=1 to n :
  for c=1 to C :
    if ( w_i > c ) :
        OPT_{i,c} = OPT_{i-1,c} ;
    else :
        OPT_{i,c} = max { OPT_{i-1,c} ; OPT_{i-1,c-w_i} + v_i }
return OPT_{n,C} ;
```

$\to T(n) = \Theta(|I| \cdot |C|)$

numero oggetti * pesi possibili.

aggiungo nella soluzione i $(\dots)$

Abbiamo bisogno di una S_{ic} → Matrice.

stampiamo tramite PRINT.

PRINT_SOL (i, c, OPT) ⟶ matrice soluzione.

```
if (i=0 ∨ c=0):
    print " "
else (i>0 ∧ c>0):
    if w_i > c :
```

```
        PRINT_SOL ( i-1, c, OPT );
  if  OPT_{i-1,c-w_i} + v_i >= OPT_{i-1,c}
        PRINT_SOL ( i-1, c-w_i, OPT );
        print( i );
else:
        PRINT_SOL ( i-1, c, OPT );
```

Complessitá del problema :

$\Theta(n \cdot \underset{|c|}{\underset{\downarrow}{m}})$ $\mapsto$ pseudo-polinomiale $\leadsto$ non é np-completo.

ESERCIZIO:

ISTANZA $n \quad (X_n = \{1 \dots n\})$

$\forall i \in \{1 \dots n\} \qquad V_i \in \mathbb{N} \quad V_i > 0$

$\qquad\qquad\qquad\qquad\qquad w_i \in \mathbb{N} \quad w_i > 0$

C = capacità zaino

$R > 0$ intero.

Ogni ogg. ha associato un colore $\quad col : \{1 \dots n\} \longrightarrow$ rosso, blu.

SOLUZIONE $S \subseteq X_n \qquad A \subseteq X_n$

$$V(S) = \max_{W(A) \leq C} \{V(A)\}$$

con # oggetti rossi in $A \leq R$

Secondo vincolo: # oggetti rossi al più R

Caso base: $\Rightarrow$ guarda meglio def. variabili negli appunti prof.

① $i = 0 \qquad X_i = \phi \qquad \forall r, c$

$S_{i,c,r} = \phi \quad \simeq \quad OPT_{i,c,r} = V(S_{i,c,r}) = 0$

② $c = 0 \qquad X_i = \phi \qquad \forall i, r$

$S_{i,c,r} = \phi \quad \simeq \quad OPT_{i,c,r} = V(S_{i,c,r}) = 0$

$\uparrow$ insieme $\qquad\qquad \uparrow$ valore associato all'insieme.

Caso passo: (i, c, r) con $i > 0$, $c > 0$, qualsiasi r

1. Se $w_i > c$ "non ci sta nello zaino"

$$S_{i,c,r} = S_{i-1,c,r}$$

2. Se $w_i \geq c$ "ci sta nello zaino"

   2a Se $i \notin S_{i,c,r}$ $\quad S_{i,c,r} = S_{i-1,c,r}$

   2b Se $i \in S_{i,c,r}$ $\leadsto$ indipendentemente da r (specializz) sotto.

      2b1 ∘ $col(i) =$ "rosso" $\wedge$ $r = 0$

      $i \notin S_{i,c,r}$ $\quad S_{i,c,r} = S_{i-1,c,r}$

      ∘ $col(i) =$ "rosso" $\wedge$ $r > 0$

      $i \in S_{i,c,r}$ $\quad S_{i,c,r} = S_{i-1,c-w_i,r-1} + v_i$

(guarda albero, finisci da solo :D).

mia sol (non finita)

Se $i=0 \vee c=0$ e R qualsiasi

$S_{i,c,r} = \emptyset$

Atrimenti :

Se $W_i > c$ e qualsiasi r

Allora $S_{i,c,r} = S_{i-1,\, c-1,\, r}$

Se $w_i \leq c \wedge Col(i) = \text{"rosso"} \wedge r=0$

Allora $S_{i,c,r} = S_{i-1,c,0}$ → cioè "primo el. blu compatibile).

Se $w_i \leq c \wedge col(i) = \text{"rosso"} \wedge r > 0$

Allora $S_{i,c,r} =$

$$\max \left\{ \underbrace{S_{i-1,c-w_i,r-1}}_{\text{Prendo solut.}} + \underbrace{V_i}_{\text{valore attuale}} ,\ \underbrace{S_{i-1,c,r}}_{\text{prendo sol. migliore con blu/rosso}} ,\ \underbrace{S_{i-1,c,r-1}}_{\text{prendo un rosso ma}} \right\}$$

Se $w_i \leq c \wedge col(i) = \text{"blu"} \wedge (r > 0 \vee r = 0)$

Allora $S_{i,c,r} =$

$$\max \left\{ S_{i-1,c-w_i,r} + V_i \ ;\ S_{i-1,c,r} \right\}$$