Zaino

Wednesday, 15 November 2023 14:43

- Dobbiamo rapinare la borsetta di una vecchietta.
  La vecchietta ha N oggetti, ed ogni oggetti ha un valore ed un peso
  - Dobbiamo rubare il maggior numero di valore dalla vecchietta
  - Senza farli notare che qualcosa è stato preso, quindi il peso totale deve essere minore di C
- Definizione problema:
  - Istanza:
    - $X = \{1, \dots, n\}$
      Dove questi sono gli oggetti
      Ed ogni oggetto ha un valore ed un peso:
      (v_i, w_i)
      $\forall j \in \{1, \dots, i\}, c \in \{0, \dots, C\}$
  - Soluzione:
    - $S\ c\ \{1, \dots, n\}\ tc$
      $$W(S_{ic}) < c \wedge V(S_{ic}) = \max_{A c \{1,\dots,i\}, W(A) \leq c} \{V(A)\}$$
      Nota:
      $$V(A) = \begin{cases} \sum A^{v_i} & A \neq \epsilon \\ 0 & A = \epsilon \end{cases}$$
      $$V: P(\{1, \dots, n\}) \to \mathbb{R}_+$$
      $$W(A) = \begin{cases} \sum i \in A^{w_i} & A \neq \epsilon \\ 0 & A = \epsilon \end{cases}$$
      $$W: P(\{1, \dots, n\}) \to \mathbb{R}_+$$
- Equazione ricorrenza
  - Caso base: i=0 v c=0
    Qui o non abbiamo più spazio oppure non c'è più nulla da rubare
    $OPT_{ic} = 0, S_{ic} = \epsilon$
  - Passi ricorsivi:
    - $w_i > c$
      Se l'oggetto ha ingombro maggiore della capacità, dobbiamo ignorarlo

$OPT_{ic} = OPT_{i-1,c}, S_{ic} = S_{i-1,c}$

- $W_i \leq c$
  L'altro caso, e qui abbiamo 2 opzioni:
  - Lo prendiamo
    Quindi fa parte di S_i
    $OPT_{ic} = OPT_{i-1,c-w_i} + v_i, S_{i,c} = S_{i-1,c} \cup \{i\}$
  - Non lo prendiamo
    Quindi non fa parte di Si
    $OPT_{ic} = OPT_{i-1,c}, S_{ic} = S_{i-1,c}$

  Prendiamo il migliore dei 2
  $OPT_{ic} = \max\{OPT_{i-1,c}, OPT_{i-1,c-w_i} + v_i\}$

Riscritto tutto meglio:

$$KS(n, c, v, w) = \begin{cases} 0 & i = 0 \ v \ C = 0 \\ KS(i-1, c, v, w) & w[i] > c \\ Giacomo^* & else \end{cases}$$

$Giacomo^* = \max\{KS(i-1, c, v, w), KS(i-1, c, v, w) + v(i)\}$

Nota: togliere o mettere v, w è indifferente

- Esempio pratico

i	v_i	w_i
1	1	1
2	6	2
3	18	5
4	22	6
5	28	7

Abbiamo C=11
  - Abbiamo presi S={3, 4}, V(S)=40
- Pseudocodice
  - Ricorsivo

```
KPric(i, c):
        If i=0 v c=0:
                Return (0, ϵ)
        Else:
                If w_i > c:
                        KPric(i-1, c)
                Else:
                        (OPT1, S1) = KPric(i-1, c)
                        (OPT2, S2) = KPric(i-1, c-wi)
                        OPT2 = OPT2 + vi
                        If OPT1 >= OPT2:
                                Return (V1, S1)
                        Else:
                                Return (V2, S2 u {i})
```

  - Bottom-up

```
KP(n, C):
    If i==0 or C==0:
        Return (0, ϵ)

    For i=0 to n:
        OPT[i, 0] = 0
        S[i, 0] = {}
    For c=0 to C:
        OPT[1, c] = 0
        S[1, c] = {}

    For i=1 to n:
        If wi > c:
            OPT[i, c] = OPT[i-1, c]
            S[i, c] = S[i-1, c]
        Else:
            V1 = OPT[i-1, c]
            V2 = OPT[i-1, c-wi] + vi
            If V1 >= V2:
                S[i, c] = S[i-1, c]
                OPT[i, c] = V1
            Else:
                S[i, c] = S[i-1, c-wi] u {i}
    Return OPT[n, C], S[n, C]
```