WIS

Monday, 9 October 2023 08:52

- Sia A un insieme di n attività
    - $A = \{1, 2, \dots, n\}$
    - $n \in N$

  Ad ogni attività $i \in A$ sono associate:
    - s_i → Tempo inizio attività
    - $e_i, e_i > s_i$ → Tempo fine attività
    - u_i → Il valore dell'attività

  E si definiscono le seguenti funzioni:
    - COMP

      compatibilità, date 2 attività $i, j \in A$ si dice che le due attività se esse non si sovrappongono:

      $[s_i, e_i) \cap (s_j, e_j] = \epsilon$
        - Si dice che un insieme A contiene attività mutualmente compatibili sse

          $\forall i, j \in A, i \neq j, [s_i, e_i) \cap (s_j, e_j] = \epsilon \mid \rightarrow Giovanni^*$

          E noi diciamo che la funzione COMP dice se A è questo oppure no

          $$COMP(A) = \begin{cases} true & Giovanni^* \\ False & else \end{cases}$$
    - V

      Associa ad ogni sottoinsieme di attività A la sua somma di valori complessivi

      $$V(A) = \begin{cases} \sum_{i \in A}^{i=0} A^{v_i} & A \neq \epsilon \\ 0 & A = \epsilon \end{cases}$$
    - v^-

      Questa funzione ci ritorna un attività j

      Dove

      $v^- = \max\{j \mid j < i, Giovanni^*(i, j)\}$

      Ritorna un attività compatibile di i e che inizia prima di j
- Definizione sottoproblema

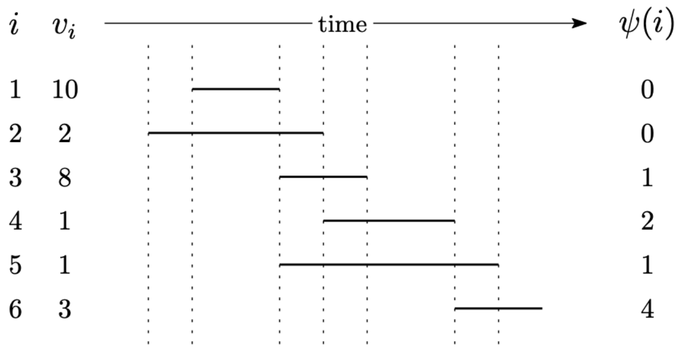
  Dobbiamo trovare un sottoinsieme di attività che sono mutualmente compatibili e di valore massimo

- Istanza: $\{1, \dots, n\}$ con $([s_i, e_j), v <_i) \forall i \in \{1, \dots, n\}$
- Soluzione: $S\ c\ \{1, \dots, n\}\ tc$

$$COMP(S) = True\ ^\wedge\ V(S) = \max_{Ac\{1,\dots,n\}:COMP(A)=True} \{V(A)\}$$

Il sottoproblema è definito solamente da $\{1, \dots, n\}$

- Esempio



La notra soluzione ottimale sarebbe:
$S = \{1,3,6\}$
Siccome hanno valore massimo

- Risoluzione
  - Caso base: $i = 0$
    Quando i=0 nn abbiamo nulla, quindi
    $OPT_i = 0, S_i = \epsilon$
  - Passo ricorsivo
    Qui noi abbiamo: A insime di attività
    E scorriamo su di a da i=0 a n
    Si suppone che le attività siano ordinate in tempo di fine
    Noi abbiamo 2 casi:
    - Prendiamo i
      Quindi $OPT_i = OPT_{v^-(i)} + v(i)$
      $S_i = S_{v^-(i)} \cup \{i\}$
    - Non prendiamo i
      Quindi $OPT_i = OPT_{i-1}$
      $S_i = S_{i-1}$

    E poi dobbiamo prendere il massimo dei 2.

$$S_i = \begin{cases} S_i & OPT_{i-1} \geq OPT_{v^-(i)} + v(i) \\ S_{v^-(i) \cup \{i\}} & else \end{cases}$$

- Pseudocodice ricorsione

```
WISRic(i):
    If i = 0:
        Return (0, ϵ)
    Else:
        OPT1, S1 = WISRic(i-1)
```

OPT2, S2 = WISRic($v^{-}(i)$)
If OPT1 > OPT2:
Return (OPT1, S1)
Else:
Return (OPT2, S2)

- Pseudocodice bottom-up

```
WIS(n):
    OPT[0] = 0
    S = []
    For i=1 to n:
        OPT1 = OPT[i-1]
        OPT2 = OPT[v⁻(i)]
        If OPT1 >= OPT2:
            S[i] = S[i-1]
            OPT[i] = V1
        Else:
            S[i] = S[v⁻(i)] u {i}
    Return (OPT[n], S[n])
```

- Stampa
  - Pseudocodice

```
Print(i):
If i!=0:
    If OPT[i] >= OPT[i-1]:
        Print(i-1)
        Printina(i)
    Else:
        Print(i-1)
```