Algoritmi e Strutture Dati (Complementi)

Appello del 30/06/2003 – Prof. Giancarlo Mauri

Soluzioni

Esercizio 1.

Marco e Alessandro devono dividersi equamente un'eredità. L'eredità consiste di n oggetti, il cui valore commerciale è rispettivamente uguale a $v_1, v_2, \ldots, v_n$ euro. Ogni oggetto è indivisibile, e va quindi dato interamente a Marco oppure ad Alessandro. Affinché la suddivisione sia equa, la somma del valore commerciale degli oggetti presi da Marco deve essere uguale alla somma del valore commerciale degli oggetti presi da Alessandro. Inoltre, nessun oggetto deve rimanere non assegnato: ovvero, ogni oggetto deve andare o a Marco o ad Alessandro.

Tuttavia, né Marco né Alessandro sono in grado di dire se una tale suddivisione sia possibile, e quindi vorrebbero utilizzare un algoritmo che dica loro se esiste una suddivisione equa dell'eredità.

Progettate un algoritmo di programmazione dinamica che consenta loro di risolvere il problema, scrivendo prima le equazioni di ricorrenza che esprimono una soluzione ottima e poi lo pseudo–codice dell'algoritmo. Infine, si dica qual è la complessità in tempo dell'algoritmo proposto.

Soluzione. Il problema può essere formalizzato come segue: dato un insieme $V = \{v_1, v_2, \ldots, v_n\}$ di interi positivi, esiste un sottoinsieme V' di V tale che $\sum_{v \in V'} v = \sum_{v \in V \setminus V'} v$?

In altre parole, posto $m = \sum_{i=1}^{n} v_i$, esiste $V' \subseteq V$ tale che $\sum_{v \in V'} v = \frac{m}{2}$?

Osserviamo anzitutto che se m è dispari possiamo immediatamente rispondere NO. Se invece m è pari, definiamo una matrice $M[1..n, 1..\frac{m}{2}]$ a valori in $\{0, 1\}$, dove per $i \in \{1, \ldots, n\}$ e $j \in \{1, \ldots, \frac{m}{2}\}$ vale $M[i, j] = 1$ se e solo se esiste un sottoinsieme A di $\{v_1, \ldots, v_i\}$ tale che $\sum_{a \in A} a = j$. Calcolati i valori di M, risponderemo SI se $M[n, \frac{m}{2}] = 1$, e NO altrimenti. I valori di M verranno calcolati per righe, a partire dalla prima; all'interno di ogni riga, i valori verranno calcolati da sinistra verso destra (e quindi, in particolare, l'elemento $M[n, \frac{m}{2}]$ sarà l'ultimo ad essere calcolato).

Per calcolare $M[i, j]$, osserviamo che affinché sia $M[i, j] = 1$ ci sono due possibilità:

1. esiste un insieme $A \subseteq \{v_1, \ldots, v_i\}$ tale che $\sum_{a \in A} a = j$ e $v_i \in A$, oppure
2. v_i *non* appartiene a *nessun* insieme $A \subseteq \{v_1, \ldots, v_i\}$ tale che $\sum_{a \in A} a = j$.

Nel primo caso deve esistere un insieme $B \subseteq \{v_1, \ldots, v_{i-1}\}$ tale che $\sum_{b \in B} b = j - v_i$, e quindi deve valere $M[i-1, j-v_i] = 1$ (si noti che deve essere $i > 1$ e $j > v_i$, altrimenti l'elemento $M[i-1, j-v_i]$ non esiste). Nel secondo caso deve esistere un insieme $B \subseteq \{v_1, \ldots, v_{i-1}\}$ tale che $\sum_{b \in B} b = j$, e quindi deve valere $M[i-1, j] = 1$ (sempre con la richiesta che valga $i > 1$).

Pertanto, le equazioni di ricorrenza che consentono di calcolare il valore di $M[i, j]$ sono:

$$M[i,j] = \begin{cases} \max\{M[i-1, j-v_i], M[i-1, j]\} & \text{se } i > 1 \text{ e } j > v_i \\ M[i-1, j] & \text{se } i > 1 \text{ e } j < v_i \\ 1 & \text{se } j = v_i \\ 0 & \text{se } i = 1 \text{ e } v_1 \neq j \end{cases}$$

mentre lo pseudocodice dell'algoritmo che calcola i valori della matrice M è il seguente:

```
PARTITION({v_1, v_2, ..., v_n})
m ← Σ_{i=1}^{n} v_i
if m mod 2 = 1
   then result ← 0
   else for j ← 1 to m/2
            do M[1, j] ← 0
        M[1, v_1] ← 1
        for i ← 2 to n
            do for j ← 1 to m/2
                   do if j = v_i
                         then M[i, j] ← 1
                         else M[i, j] ← M[i − 1, j]
                              if j > v_i and M[i − 1, j − v_i] > M[i, j]
                                 then M[i, j] ← M[i − 1, j − v_i]
        result ← M[n, m/2]
return result
```

L'algoritmo restituisce 1 se è possibile spartire l'eredità in modo equo, altrimenti restituisce 0. Il tempo di calcolo (nel caso peggiore) è $\Theta(n \cdot \frac{m}{2}) = \Theta(n \cdot m)$.

Esercizio 2.

Sia $S = \{1, \ldots, 1000\}$ l'insieme dei primi 1000 interi positivi, e sia I la famiglia di sottoinsiemi di S tale che $A \subseteq S$ appartiene a I se e solo se:

$$\sum_{a \in A} a \equiv 0 \pmod 3$$

ovvero se e solo se la somma di tutti gli elementi di A è divisibile per 3.

- La coppia (S, I) è un sistema di indipendenza?
- La coppia (S, I) è un matroide?

Per ciascuna delle due domande dare una dimostrazione se la risposta è affermativa, oppure un controesempio se la risposta è negativa.

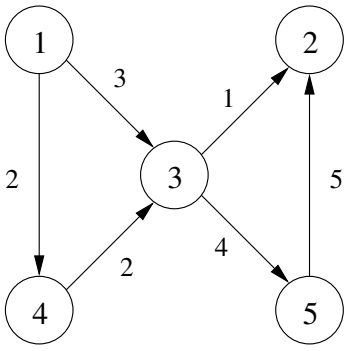
Soluzione. La coppia (S, I) proposta non è un sistema di indipendenza. Per esserlo, infatti, dovrebbe valere la **proprietà di ereditarietà**:

$$A \in I \quad \text{e} \quad B \subseteq A \quad \Longrightarrow \quad B \in I$$

Se poniamo $A = \{1, 2\}$ e $B = \{1\}$ allora valgono $A \in I$ e $B \subseteq A$, ma chiaramente $B \notin I$. Essendo violata la proprietà di ereditarietà, la coppia (S, I) proposta non è un sistema di indipendenza.

Per lo stesso motivo la coppia (S, I) non può essere un matroide. Per esserlo, infatti, dovrebbe essere un sistema di indipendenza e inoltre dovrebbe valere la **proprietà dello scambio**. Non essendo un sistema di indipendenza, possiamo immediatamente concludere che la coppia (S, I) proposta non è un matroide.

Esercizio 3.
Mostrare la sequenza delle matrici $D^{(k)}$ e $\Pi^{(k)}$ calcolate dall'algoritmo di Floyd–Warshall per il grafo indicato in figura:



Soluzione. Ricordiamo che l'algoritmo di Floyd–Warshall è un algoritmo di programmazione dinamica che calcola i cammini minimi tra tutte le coppie di vertici di un grafo orientato e pesato. Il grafo dato in ingresso può avere archi di peso negativo, ma è necessario che non vi siano cicli di peso negativo (osserviamo che il grafo proposto verifica le condizioni imposte dall'algoritmo). Detto n il numero di vertici del grafo, il tempo di esecuzione dell'algoritmo di Floyd–Warshall è $\Theta(n^3)$.

Per $i, j \in \{1, 2, \ldots, n\}$ e $k \in \{0, 1, \ldots, n\}$, indichiamo con $d_{ij}^{(k)}$ l'elemento di posto (i, j) della matrice $D^{(k)}$, ovvero la distanza di un cammino minimo tra il vertice i e il vertice j avente tutti i vertici intermedi nell'insieme $\{1, 2, \ldots, k\}$. Per il grafo proposto, l'algoritmo di Floyd–Warshall calcola le matrici $D^{(0)}$, $D^{(1)}$, $D^{(2)}$, $D^{(3)}$, $D^{(4)}$, $D^{(5)}$, dove:

$$d_{ij}^{(0)} = \begin{cases} 0 & \text{se } i = j \\ \text{il peso dell'arco orientato } (i,j) & \text{se } i \neq j \text{ e } (i,j) \in E \\ \infty & \text{se } i \neq j \text{ e } (i,j) \notin E \end{cases}$$

mentre per $k \in \{1, \ldots, 5\}$ la matrice $D^{(k)}$ viene calcolata a partire da $D^{(k-1)}$ utilizzando la seguente relazione:

$$d_{ij}^{(k)} = \min\left\{d_{ij}^{(k-1)},\ d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right\}$$

Analogamente, per $i, j \in \{1, 2, \ldots, n\}$ e $k \in \{0, 1, \ldots, n\}$ indichiamo con $\pi_{ij}^{(k)}$ l'elemento di posto (i, j) della matrice $\Pi^{(k)}$, ovvero il predecessore del vertice j su un cammino minimo dal vertice i avente tutti i vertici intermedi nell'insieme $\{1, 2, \ldots, k\}$. Per il grafo proposto, l'algoritmo di Floyd–Warshall calcola le matrici $\Pi^{(0)}$, $\Pi^{(1)}$, $\Pi^{(2)}$, $\Pi^{(3)}$, $\Pi^{(4)}$, $\Pi^{(5)}$, dove:

$$\pi_{ij}^{(0)} = \begin{cases} \text{NIL} & \text{se } i = j \text{ oppure } w_{ij} = \infty \\ i & \text{se } i \neq j \text{ e } w_{ij} < \infty \end{cases}$$

mentre per $k \in \{1, \ldots, 5\}$ la matrice $\Pi^{(k)}$ viene calcolata a partire da $\Pi^{(k-1)}$ e $D^{(k-1)}$ utilizzando la seguente relazione:

$$\pi_{ij}^{(k)} = \begin{cases} \pi_{ij}^{(k-1)} & \text{se } d_{ij}^{(k-1)} \leq d_{ik}^{(k-1)} + d_{kj}^{(k-1)} \\ \pi_{kj}^{(k-1)} & \text{se } d_{ij}^{(k-1)} > d_{ik}^{(k-1)} + d_{kj}^{(k-1)} \end{cases}$$

Pertanto, le matrici $D^{(0)}$, $D^{(1)}$, $D^{(2)}$, $D^{(3)}$, $D^{(4)}$, $D^{(5)}$, e le corrispondenti matrici $\Pi^{(0)}$, $\Pi^{(1)}$, $\Pi^{(2)}$, $\Pi^{(3)}$, $\Pi^{(4)}$, $\Pi^{(5)}$ calcolate dall'algoritmo di Floyd–Warshall sono le seguenti:

$$D^{(0)} = \begin{pmatrix} 0 & \infty & 3 & 2 & \infty \\ \infty & 0 & \infty & \infty & \infty \\ \infty & 1 & 0 & \infty & 4 \\ \infty & \infty & 2 & 0 & \infty \\ \infty & 5 & \infty & \infty & 0 \end{pmatrix} \qquad \Pi^{(0)} = \begin{pmatrix} \text{NIL} & \text{NIL} & 1 & 1 & \text{NIL} \\ \text{NIL} & \text{NIL} & \text{NIL} & \text{NIL} & \text{NIL} \\ \text{NIL} & 3 & \text{NIL} & \text{NIL} & 3 \\ \text{NIL} & \text{NIL} & 4 & \text{NIL} & \text{NIL} \\ \text{NIL} & 5 & \text{NIL} & \text{NIL} & \text{NIL} \end{pmatrix}$$

$$D^{(1)} = \begin{pmatrix} 0 & \infty & 3 & 2 & \infty \\ \infty & 0 & \infty & \infty & \infty \\ \infty & 1 & 0 & \infty & 4 \\ \infty & \infty & 2 & 0 & \infty \\ \infty & 5 & \infty & \infty & 0 \end{pmatrix} \qquad \Pi^{(1)} = \begin{pmatrix} \text{NIL} & \text{NIL} & 1 & 1 & \text{NIL} \\ \text{NIL} & \text{NIL} & \text{NIL} & \text{NIL} & \text{NIL} \\ \text{NIL} & 3 & \text{NIL} & \text{NIL} & 3 \\ \text{NIL} & \text{NIL} & 4 & \text{NIL} & \text{NIL} \\ \text{NIL} & 5 & \text{NIL} & \text{NIL} & \text{NIL} \end{pmatrix}$$

$$D^{(2)} = \begin{pmatrix} 0 & \infty & 3 & 2 & \infty \\ \infty & 0 & \infty & \infty & \infty \\ \infty & 1 & 0 & \infty & 4 \\ \infty & \infty & 2 & 0 & \infty \\ \infty & 5 & \infty & \infty & 0 \end{pmatrix} \qquad \Pi^{(2)} = \begin{pmatrix} \text{NIL} & \text{NIL} & 1 & 1 & \text{NIL} \\ \text{NIL} & \text{NIL} & \text{NIL} & \text{NIL} & \text{NIL} \\ \text{NIL} & 3 & \text{NIL} & \text{NIL} & 3 \\ \text{NIL} & \text{NIL} & 4 & \text{NIL} & \text{NIL} \\ \text{NIL} & 5 & \text{NIL} & \text{NIL} & \text{NIL} \end{pmatrix}$$

$$D^{(3)} = \begin{pmatrix} 0 & 4 & 3 & 2 & 7 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & 1 & 0 & \infty & 4 \\ \infty & 3 & 2 & 0 & 6 \\ \infty & 5 & \infty & \infty & 0 \end{pmatrix} \qquad \Pi^{(3)} = \begin{pmatrix} \text{NIL} & 3 & 1 & 1 & 3 \\ \text{NIL} & \text{NIL} & \text{NIL} & \text{NIL} & \text{NIL} \\ \text{NIL} & 3 & \text{NIL} & \text{NIL} & 3 \\ \text{NIL} & 3 & 4 & \text{NIL} & 3 \\ \text{NIL} & 5 & \text{NIL} & \text{NIL} & \text{NIL} \end{pmatrix}$$

$$D^{(4)} = \begin{pmatrix} 0 & 4 & 3 & 2 & 7 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & 1 & 0 & \infty & 4 \\ \infty & 3 & 2 & 0 & 6 \\ \infty & 5 & \infty & \infty & 0 \end{pmatrix} \qquad \Pi^{(4)} = \begin{pmatrix} \text{NIL} & 3 & 1 & 1 & 3 \\ \text{NIL} & \text{NIL} & \text{NIL} & \text{NIL} & \text{NIL} \\ \text{NIL} & 3 & \text{NIL} & \text{NIL} & 3 \\ \text{NIL} & 3 & 4 & \text{NIL} & 3 \\ \text{NIL} & 5 & \text{NIL} & \text{NIL} & \text{NIL} \end{pmatrix}$$

$$D^{(5)} = \begin{pmatrix} 0 & 4 & 3 & 2 & 7 \\ \infty & 0 & \infty & \infty & \infty \\ \infty & 1 & 0 & \infty & 4 \\ \infty & 3 & 2 & 0 & 6 \\ \infty & 5 & \infty & \infty & 0 \end{pmatrix} \qquad \Pi^{(5)} = \begin{pmatrix} \text{NIL} & 3 & 1 & 1 & 3 \\ \text{NIL} & \text{NIL} & \text{NIL} & \text{NIL} & \text{NIL} \\ \text{NIL} & 3 & \text{NIL} & \text{NIL} & 3 \\ \text{NIL} & 3 & 4 & \text{NIL} & 3 \\ \text{NIL} & 5 & \text{NIL} & \text{NIL} & \text{NIL} \end{pmatrix}$$