Prova scritta del 2 febbraio 2018 – Nota sulla soluzione dell'esercizio 4

Premessa. Il problema proposto è una variante del problema del calcolo di un cammino di valore minimo su una matrice, presentato e risolto nella lezione del 4 dicembre 2017 (si veda anche il paragrafo 10.2 del libro di testo). La soluzione dell'esercizio può essere ottenuta con semplici modifiche alla soluzione di tale problema. Quella che segue NON è la soluzione completa dell'esercizio, ma una nota con indicazioni su come si può procedere, insieme a uno schema della soluzione.

(a) Per calcolare in valori $c_{i,j}$ sulla base dei valori dei cammini che terminano nella colonna $j-1$, studiamo come la casella (i,j) possa essere raggiunta, in una mossa, da una casella della colonna $j-1$.

In base ai vincoli sulle mosse, se la casella è vuota, essa può essere raggiunta solo dalla casella immediatamente alla sua sinistra. Inoltre, l'ingresso in una casella vuota non modifica il punteggio del giocatore. Pertanto, se $m_{i,j}=0$ allora il valore da assegnare a $c_{i,j}$ è quello di $c_{i,j-1}$.

Una casella contenente un pezzo (quindi con $m_{i,j}>0$) può essere invece raggiunta solo dalla casella alla sua sinistra nella riga superiore o dalla casella alla sua sinistra nella riga inferiore. Inoltre, in questo caso al punteggio deve essere aggiunto il valore presente nella casella. Pertanto, al fine di massimizzare il punteggio, tra le due possibili mosse che entrano nella casella (i,j) si sceglie quella che proviene dal cammino con punteggio più altro. Dunque, in questo caso il valore da assegnare a $c_{i,j}$ è $\max\{c_{i-1,j-1}, c_{i+1,j-1}\}+m_{i,j}$.

Riassumendo, per $j>1$ e $i=1,\ldots,n$, scriviamo la seguente formula:

$$c_{i,j}=\begin{cases} c_{i,j-1} & \text{if } m_{i,j}=0,\\ \max\{c_{i-1,j-1}, c_{i+1,j-1}\}+m_{i,j} & \text{altrimenti.}\end{cases}$$

In realtà se $i=1$ la riga superiore non esiste, per cui al posto di $\max\{c_{0,j-1}, c_{2,j-1}\}+m_{1,j}$ dobbiamo scrivere semplicemente $c_{2,j-1}+m_{1,j}$. Analogamente, nel caso $i=n$ dobbiamo adattare la formula per tenere conto del fatto che non esiste la riga $n+1$. Per evitare di scrivere la formula in maniera più complicata, potremmo ipotizzare di aggiungere alla matrice C una riga 0 e una riga $n+1$ nelle quali tutti gli elementi sono uguali a 0.

La soluzione del problema è l'elemento massimo dell'ultima colonna di C, cioè $\max\{c_{i,n} \mid i=1,\ldots,n\}$.

(b) Si osservi che per come definito nel testo dell'esercizio, l'elemento $c_{i,j}$ rappresenta la soluzione del problema "trovare il punteggio massimo di un cammino che inizia in una qualunque casella della colonna di sinistra e termina nella casella (i,j)". La tecnica di programmazione dinamica calcola le soluzioni di questi problemi, memorizzandole nella matrice C e, infine, ricava da esse la soluzione del problema dato.

L'algoritmo è costituito da due parti: la prima riempie la matrice C, la seconda determina la soluzione del problema dato calcolando il massimo dell'ultima colonna di C.

Per riempire la matrice C si procede *per colonne* (infatti i valori in una colonna dipendono da quelli nella colonna precedente). La prima colonna viene riempita copiando la prima colonna di M. Infatti, come specificato nel testo dell'esercizio, $c_{i,1}=m_{i,1}$ per $i=1,\ldots,n$. Per riempire le colonne successive si scrivono due cicli innestati, secondo questo schema (si osservi che, in base a quanto già discusso, il ciclo esterno deve variare l'indice delle colonne e non quello delle righe):

```
FOR j ← 2 TO n
    FOR i ← i TO n
        qui si scrive il codice che assegna a c_{i,j} il valore
        sulla base della formula ottenuta al punto (a)
```

(c) La stima del tempo deve essere ricavata e *giustificata* sulla base dello pseudocodice scritto. In questo caso la parte più costosa del codice è quella schematizzata sopra, con due cicli innestati, all'interno dei quali si effettuano operazioni di costo complessivo $O(1)$. Pertanto il tempo risulta dell'ordine di n^2.