Corso di Algoritmi e Strutture Dati
Anno Accademico 2019–2020
Prof. Giovanni Pighizzini

Esercizi

8. Programmazione dinamica

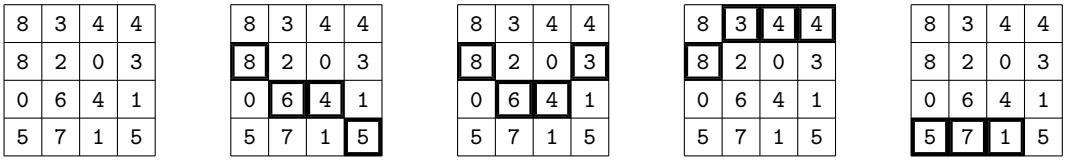
Esercizio 8.1
Progettate un algoritmo che data una matrice quadrata di n righe e n colonne determini per ogni riga r il cammino di valore minimo (come definito nel paragrafo 10.2.2 del libro di testo) che inizia dall'elemento di posizione $(r, 1)$ e termina su un qualunque elemento dell'ultima colonna. Date una stima dei tempi di calcolo dell'algoritmo ottenuto.
Suggerimento: potete ispirarvi all'esempio riportato sul libro di testo, calcolando una matrice tridimensionale C, in cui $C[r, i, j]$ contiene il minimo valore di un cammino che inizia nella posizione $(r, 1)$ e termina nella posizione (i, j).

Esercizio 8.2
Un robot si muove sulle caselle di una griglia quadrata $n \times n$ partendo da una casella della colonna più a sinistra. Il robot è alimentato con delle mele, collocate in un apposito serbatoio.

- Inizialmente nel serbatoio vengono collocate tutte le mele che si trovano nella casella della colonna di sinistra da cui il robot inizia a muoversi.
- Per spostarsi da una casella il robot ha bisogno di mangiare K mele (ciò avviene anche quando il robot si sposta da una casella per abbandonare la griglia). Se il robot ha meno di K mele rimane bloccato in attesa di soccorsi.
- Ogni volta che il robot raggiunge una casella, mette nel serbatoio tutte le mele che vi trova (per semplicità supponete che la capacità del serbatoio sia illimitata).

Il robot può scegliere arbitrariamente una casella della colonna di sinistra da cui iniziare il proprio percorso. Ad ogni spostamento il robot può muoversi esclusivamente in una delle seguenti caselle della colonna immediatamente a destra: la casella sulla stessa riga, la casella sulla riga superiore (se esiste), la casella sulla riga inferiore (se esiste). L'obiettivo è abbandonare la griglia spostandosi all'esterno da una qualunque casella della colonna di destra.

Esempio. La figura qui sotto a sinistra rappresenta una griglia con l'indicazione del numero di mele presenti in ciascuna casella. Nelle altre figure sono evidenziati quattro possibili cammini nel caso $K = 5$. Nei primi due cammini il robot riesce ad uscire a destra, negli altri rimane bloccato.



Data pubblicazione: 11 dicembre 2019

Cosa si richiede. Sono dati una matrice M di dimensione $n \times n$, contenente interi non negativi (l'elemento $m_{i,j}$ è il numero di mele presenti nella casella (i, j)) e il valore K (numero di mele necessarie per spostarsi da una casella a un'altra). Indichiamo con $c_{i,j}$ il massimo numero di mele che il robot può avere nel serbatoio seguendo un percorso che inizia in una qualunque casella della colonna di sinistra (cioè la numero 1) e termina nella casella (i, j). Nel caso il robot non sia in grado di raggiungere la casella (i, j), assegnamo a $c_{i,j}$ il valore $-\infty$. Si osservi che $c_{i,1} = m_{i,1}$, per $i = 1, \ldots, n$. Sia C la matrice dei valori $c_{i,j}$.

(a) Tenendo conto delle regole con cui il robot si può muovere, scrivete una o più formule o, in alternativa, righe di pseudocodice, che permettano di ricavare il generico valore $c_{i,j}$ della colonna j di C, con $j > 1$, sulla base di K, di alcuni valori della colonna $j - 1$ di C e del valore $m_{i,j}$.

(b) Descrivete *sinteticamente a parole* e poi *ad alto livello* in pseudocodice, un algoritmo basato sulla tecnica di *programmazione dinamica* che ricevendo in ingresso la matrice M e il valore K riempia la matrice C (utilizzate quanto scritto al punto (a)) e stabilisca se esista almeno un percorso che permetta al robot partendo da una casella nella colonna di sinistra di uscire dalla colonna di destra.

(c) Fornite una stima del tempo totale utilizzato dall'algoritmo in funzione di n.

Nota. Potete effettuare operazioni direttamente con il valore $-\infty$. Il risultato dell'addizione e della sottrazione tra $-\infty$ e un valore finito è $-\infty$. Il minimo e il massimo tra $-\infty$ e un valore x sono rispettivamente $-\infty$ e x.

Esempio.
Dati $M = \begin{bmatrix} 8 & 3 & 4 & 4 \\ 8 & 2 & 0 & 3 \\ 0 & 6 & 4 & 1 \\ 5 & 7 & 1 & 5 \end{bmatrix}$ e $K = 5$ (esempio precedente), si ottiene $C = \begin{bmatrix} 8 & 6 & 5 & 4 \\ 8 & 5 & 4 & 6 \\ 0 & 9 & 8 & 4 \\ 5 & 7 & 5 & 8 \end{bmatrix}$.
Si noti che in questo caso esiste almeno un percorso che permette al robot di uscire dalla griglia.

Per la stessa matrice M, con $K = 6$ si ottiene $C = \begin{bmatrix} 8 & 5 & -\infty & -\infty \\ 8 & 4 & 2 & 3 \\ 0 & 8 & 6 & 1 \\ 5 & -\infty & 3 & 5 \end{bmatrix}$. In questo caso non esiste alcun percorso che permette al robot di uscire dalla griglia.

Esercizio 8.3
Progettate un algoritmo che date due stringhe di caratteri determini una loro più lunga sottostringa comune. Ad esempio, se le stringhe sono `saltimbanco` e `alticcio`, il risultato dovrà essere `alti`. Date una stima dei tempi di calcolo dell'algoritmo ottenuto.
Suggerimento: potete ispirarvi all'esempio riportato sul libro di testo e visto a lezione, relativo al calcolo della distanza tra due stringhe, calcolando, prima di tutto, la massima lunghezza di una sottostringa comune.

Esercizio 8.4
Una sottosequenza di una stringa x è una qualunque stringa ottenuta da x cancellando alcuni dei suoi caratteri. Ad esempio, una sottosequenza di `saltimbanco` è `libano`. Progettate un algoritmo che date due stringhe di caratteri determini una loro più lunga sottosequenza comune. Ad esempio, se le stringhe sono `saltimbanco` e `alticcio`, il risultato dovrà essere `altico`. Date una stima dei tempi di calcolo dell'algoritmo ottenuto.