

Esercizio 2 del 16/4/2018

E' un'estensione dell'esercizio 1 di questa settimana nel quale le funzioni ricorsive partenza e trova devono fornire, in caso il cammino esista, anche il cammino che trovato, usando a questo scopo l'array P che viene loro passato come parametro (e che è dichiarato nel main). Il seguente esempio spiega come rappresentare un cammino con un array di interi.

Esempio. Supponiamo che l'input letto dal main (dato) sia il seguente:

```
6
0 0 0 1 0 1
1 0 1 0 1 0
0 0 0 0 1
0 0 0 0 1 0
0 0 0 1 0 1
1 1 1 1 1 0
```

Il cammino più a sinistra che traversa l'intera matrice dalla prima all'ultima riga è: (0,3) (1,4) (2,5) (3,4) (4,3) (5,2). Esiste anche il cammino (0,5) (1,4) (2,5) (3,4) (4,3) (5,2), ma è più a destra del precedente visto che sulla prima riga parte da una colonna (5) maggiore della colonna usata dal primo cammino (3). Allo stesso modo il cammino, (0,3) (1,4) (2,5) (3,4) (4,3) (5,3), porta dalla prima all'ultima riga, ma di nuovo la colonna usata all'ultimo passo (3) è maggiore di quella del primo cammino (2). Quindi il primo cammino è più a sinistra di questo.

Osservando i 3 cammini è facile accorgersi che le prime componenti delle loro caselle sono sempre 0...5. Quindi la prima componente è inutile e possiamo ricordarci il cammino semplicemente memorizzando in P le seconde componenti delle diverse caselle che lo compongono. Il cammino più a sinistra viene memorizzato dentro l'array P come, $P=[3,4,5,4,3,2]$. La prima componente è semplicemente l'indice in P in cui si trova la seconda componente.

Esercizio: in pratica si tratta di modificare la soluzione dell'esercizio 1 di questa settimana in modo che le funzioni partenza e trova soddisfino le seguenti specifiche:

PRE_p=(N ha $m*m$ elementi definiti, $0 \leq i \leq m$, P ha m posti)

bool partenza(bool*N, int m, int i, int*P)

POST=(restituisce true sse esiste un cammino in N, visto come $X[m][m]$, dalla riga i all'ultima riga) &&(se restituisce true allora $P[i..m-1]$ contiene il cammino più a sinistra dalla riga i alla riga m-1)

Pre_t=(N ha $m*m$ elementi definiti, $0 \leq i \leq m$, P ha m posti)

bool trova(bool*N, int m, int i, int j, int*P)

POST=(restituisce true sse esiste un cammino in N, visto come $X[m][m]$, dalla casella (i,j) all'ultima riga) &&(se restituisce true allora $P[i..m-1]$ contiene il cammino più a sinistra dalla casella (i,j) alla riga m-

Si chiede anche di definire la funzione ricorsiva stampa che stampa il cammino trovato (quando c'è). La funzione ricorsiva stampa deve obbedire alla seguente specifica:

PRE=(P ha m posizioni)

void stampa(int*P, int m, int i)

POST=(stampa nel modo indicato nel seguente esempio il cammino contenuto in $P[i..m-1]$)

Esempio: si consideri il cammino $P=[3,4,5,4,3,2]$ visto nell'esempio precedente, allora la funzione ricorsiva stampa deve produrre il seguente output: esiste un cammino e quello più a sinistra e':

(0,3) (1,4) (2,5) (3,4) (4,3) (5,2).

Correttezza:

- 1) Dimostrare induttivamente la correttezza di trova
- 2) Dimostrare induttivamente la correttezza di stampa.