

Esercizio 2 del 12/5/2017

E' un'estensione dell'esercizio 1 nel quale la funzione ricorsiva trova deve fornire, in caso il cammino esista, anche il cammino che ha trovato, usando per questo l'array P che le viene passato come parametro. La nuova specifica di trova è la seguente:

PRE=(L è un array $n \times n$, $0 \leq e \leq n$, P ha n posizioni)

bool trova(int*L, int n, int e, int*P)

POST=(restituisce true sse esiste un cammino dalla riga 0 alla riga n-1 di L, visto come array $\text{int}[n][n]$, con inizio da un elemento e..n-1 della riga 0) &&(se restituisce true, allora P descrive il cammino più a sinistra che esiste, precisamente, P è tale che per ogni i in $[0..n-1]$, P[i] contiene l'elemento sulla riga i del cammino più a sinistra che esiste)

Oltre alla funzione trova (che avrà bisogno di una funzione ricorsiva ausiliaria), si chiede 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 n posizioni)

void stampa(int*P, int n, int c)

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

Il seguente esempio spiega cosa deve fare stampa in un caso in cui un cammino esiste. Se nessun cammino esiste il main stampa semplicemente "il cammino non esiste".

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 0 1
0 0 0 0 1 0
0 0 0 1 0 1
1 1 1 1 1 0
```

La funzione stampa deve produrre il seguente output:

il cammino e': (0,3) (1,4) (2,5) (3,4) (4,3) (5,2)

Si osservi che sono presenti vari cammini e che viene stampato quello più a sinistra. Che è poi il primo che viene trovato se scrivete il programma nel modo più semplice.