

Esercizio 3 del 12/5/2017

Si tratta di modificare il programma dell'esercizio 2 (di questa settimana) in modo che la funzione ricorsiva trova soddisfi la seguente specifica:

PRE=(L ha $n \times n$ posizioni, $0 \leq e \leq n$)

nodo* trova(int*L, int n, int e)

POST=(restituisce una lista non vuota (cioè diversa da 0) se 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 una lista non vuota X , allora questa lista descrive il cammino più a sinistra che esiste, precisamente, è tale che per ogni i in $[0..n-1]$, il nodo i -esimo di X contiene l'elemento sulla riga i del cammino più a sinistra che esiste)

La lista di cui parliamo, è costituita da nodi definiti dalla seguente struttura:

```
struct nodo{int info; nodo* next;} P[rig]=col; P[rig]=col;
```

Potete introdurre nella struttura un costruttore per inizializzare i nodi in modo più semplice.

Si chiede anche di trasformare la funzione ricorsiva stampa dell'Esercizio 2 in modo che stampi (nello stesso modo dell'Esercizio 2) il cammino contenuto nella lista restituita da trova.

L'esempio dell'Esercizio 2 vale anche per questo esercizio.