

## Esame scritto di Programmazione del 13/7/2017

L'esercizio richiede di smembrare una lista concatenata in due liste concatenate, secondo certi criteri che sono spiegati nel seguito. Abbiamo una lista concatenata L costituita dai soliti nodi con un campo info int ed una seconda lista D i cui nodi contengono 2 interi non negativi. I nodi di D descrivono il modo di smembrare L. Vediamo un esempio.

**Esempio 1.** Sia  $L = 0 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow 100 \rightarrow 0 \rightarrow 200 \rightarrow 100 \rightarrow 11 \rightarrow 5 \rightarrow 2$  e sia  $D = [0,2] \rightarrow [1,3] \rightarrow [3,1] \rightarrow [0,2]$ . Le coppie di D vanno interpretate nel modo seguente, per una coppia  $[x,y]$  di D, dobbiamo **lasciare** x nodi di L e **togliere** i successivi y nodi di L. Eseguendo in questo modo quanto prescritto dai nodi di D, da L produrremo 2 liste, quella dei nodi di L che sono **lasciati** e quella dei nodi di L che sono **tolti**. Con L e D dati prima, avremo che **lasciati** =  $2 \rightarrow 200 \rightarrow 100 \rightarrow 11$ , mentre **tolti** =  $0 \rightarrow 3 \rightarrow 4 \rightarrow 100 \rightarrow 0 \rightarrow 5 \rightarrow 2$ . Le 2 liste sono formate come segue: il primo nodo  $[0,2]$  di D richiede di lasciare 0 nodi di L e di togliere i primi 2 nodi, quindi  $0 \rightarrow 3$  che infatti sono i primi 2 nodi di **tolti**. Quindi L è ora  $2 \rightarrow 4 \rightarrow 100 \rightarrow 0 \rightarrow 200 \rightarrow 100 \rightarrow 11 \rightarrow 5 \rightarrow 2$ . Su questa lista il secondo nodo  $[1,3]$  di D ha l'effetto di lasciare il primo nodo, che contiene 2, (che infatti è il primo e per ora unico nodo di **lasciati**) e di togliere i successivi tre nodi, che sono  $4 \rightarrow 100 \rightarrow 0$ . Quindi **tolti** diventa  $0 \rightarrow 3 \rightarrow 4 \rightarrow 100 \rightarrow 0$ , mentre a questo punto  $L = 200 \rightarrow 100 \rightarrow 11 \rightarrow 5 \rightarrow 2$ . Il prossimo nodo  $[3,1]$  di D aggiunge i prossimi tre nodi di L a **lasciati** e il quarto nodo a **tolti** e infine  $[0,2]$  non contribuisce per nulla a **lasciati** e aggiunge a **tolti** l'ultimo nodo 2. In realtà  $[0,2]$  richiederebbe di aggiungere a **tolti** 2 nodi, ma, visto che a questo punto in L ne resta 1 solo, quest'unico nodo presente viene aggiunto a **tolti**. Può succedere anche che D non esaurisca i nodi di L. Per esempio, se  $D = [3,2]$ , allora **lasciati** sarà  $0 \rightarrow 3 \rightarrow 2 \rightarrow 0 \rightarrow 200 \rightarrow 100 \rightarrow 11 \rightarrow 5 \rightarrow 2$ , mentre **tolti** =  $4 \rightarrow 100$ . Insomma, la parte di L che resta dopo aver esaurito D, viene inserita in **lasciati**.

**Esercizio iterativo (vale 9 punti):** scrivere una funzione **iterativa** Fiter che data una lista L ed una lista D produca le 2 liste **lasciati** e **tolti**, come descritto in Esempio 1. Fiter deve soddisfare la seguente specifica:

PRE=(Lista(L), Lista(D), Lista(lasciati) e Lista(tolti) sono corrette,  $vL=L$ ,  $vlasciati=lasciati$  e  $vtolti=tolti$ )

Fiter(nodo\* L, nodoD\* D, nodo\*&lasciati, nodo\*&tolti);

POST=(lasciati e tolti sono vlasciati e vtolti a cui sono state appese (alla fine) le liste ottenute dalla smembramento di vL come indicato dai nodi di D)

**(vale 2 punti)** Fiter **deve** usare la seguente funzione **iterativa** nodo\* taglia(nodo\*&L, int n) che soddisfa le seguenti PRE e POST:

PRE=(Lista(L) è corretta,  $n \geq 0$ ,  $vL=L$ )

POST=(la funzione restituisce col return i primi n nodi di vL se ci sono e altrimenti restituisce quello che c'è, L diventa quello che resta di vL una volta tolto il prefisso restituito col return)

Per Fiter può essere utile anche una funzione di concatenazione che viene fornita nella versione iterativa.

**Esercizio ricorsivo (vale 9 punti):** scrivere una funzione **ricorsiva** doppioN Fric(nodo\* L, nodoD\* D) che deve soddisfare la seguente coppia di condizioni:

PRE=(Lista(L) e Lista(D) sono corrette,  $vL=L$ )

POST=(restituisce col return un valore struttura doppioN che è una coppia di valori nodo\* che sono le liste **lasciati** e **tolti** (rispettivamente, nel campo La e To del valore doppioN) in cui viene smembrata vL secondo quanto prescritto da D)

La struttura doppioN è dichiarata nel programma che viene dato.

**(vale 2 punti)** La funzione Fric **deve** usare una funzione **ricorsiva** nodo\* tagliaric(nodo\*&L, int n) che soddisfa le stesse PRE e POST di taglia.

**(vale 2 punti)** Fric deve fare operazioni di concatenazione tra liste e si chiede quindi di definire una funzione ricorsiva, nodo\* concric(nodo\*L1, nodo\*L2), che restituisca col return la concatenazione di L1 ed L2.

In mancanza di tempo si può usare la conc iterativa data.

**Si osservi** che l'output prodotto da Fiter e da Fric è uguale (vedi main dato) e quindi è possibile passare tutti i test con 1 sola delle 2. Osservate infine che la lista L costruita da build (che è data) viene clonata nel main nella lista L1 (la funzione clone è data), visto che Fiter smembra L e quindi non sarebbe possibile invocare Fric passandole L che è appena stata smembrata.

**Correttezza (2 punti per ciascuna domanda)**

- 1) Dare l'invariante del ciclo principale di Fiter.
- 2) Fornire la prova induttiva di Fric.
- 3) Dare la prova induttiva di tagliaric.