

Esame del 21/6/2021 Programmazione ricorsiva

inizio 11e30 circa, 1 ora di tempo + 10 min per la consegna

Data una lista concatenata L con nodi che hanno campo info intero, vogliamo staccare i nodi da L e con essi vogliamo creare tante nuove liste quanti sono i valori distinti dei campi info dei nodi di L. Insomma per ogni valore z che appare in un campo info di un nodo di L, vogliamo creare una lista che consiste di tutti i nodi di L con info=z. In ciascuna di queste liste vogliamo che i nodi siano ordinati con lo stesso ordine che avevano in L.

Esempio 1: sia L la seguente lista: 3->1->3->2->1->3->2->3, allora si vuole ottenere 3 liste: una con i 5 nodi con info=3 di L, la seconda con i 2 nodi con info=1 di L e la terza con i 2 nodi con info=2 di L. Le 3 liste verranno mantenute in una lista di nodi di tipo nodoE che segue:

```
struct nodoE{ nodo* info; nodoE* next; nodoE(nodo* a=0, nodoE* b=0){ info=a; next=b; };
```

quindi vogliamo costruire una lista X di 3 nodi di tipo nodoE, dove ogni nodo punta ad una delle 3 liste di nodi di Lista(L) descritte prima. Inoltre vogliamo che in X le tre liste siano ordinate per campi info. Quindi il primo nodo di X deve puntare alla lista con 2 nodi con info=1, il secondo nodo di X punta alla lista con 2 nodi con info=2 e il terzo nodo punta alla lista con i 5 nodi con info=3. Nel seguito diremo che una tale lista X è ordinata.

Esempio 2: descriviamo come la lista X dovrebbe venire costruita. Inizialmente X è vuota. Consideriamo il primo nodo di Lista(L) che ha info=3, stacciamo il nodo da L e creiamo un primo nodoE di X che punta alla lista che consiste del nodo appena staccato da L. Procediamo quindi col secondo nodo di Lista(L) che ha info=1. Visto che il suo info è minore di quello del nodo già presente in X, si dovrà aggiungere un nodo all'inizio di X che col campo info punta al nodo di L con info=1 e col campo next punta al nodoE già presente in X. Poi si stacca il terzo nodo di Lista(L) che ha info=3. Poiché in X il secondo nodo punta già ad un nodo proveniente da L con info=3, si deve aggiungere il nodo appena staccato da L a questa lista puntata dal secondo nodo di X. La procedura deve continuare nello stesso modo finché tutti i nodi L non siano stati inseriti in X.

Esercizio 1 (18 punti): si scriva una funzione **ricorsiva** nodoE* insOrd(nodoE* X, nodo* y) che soddisfa la seguente PRE e POST:

PRE=(Lista(X) è ben formata e ordinata, y punta ad un nodo con campo next a 0, vX=Lista(X))

POST=(la funzione restituisce col return la lista ordinata ottenuta da vX aggiungendo ad essa il nodo y nel modo descritto in Esempio 2)

Esercizio 2 (7 punti): si scriva una funzione **ricorsiva** nodoE* SRic(nodo* L) che soddisfa la seguente PRE e POST:

PRE=(Lista(L) ben formata, vL=Lista(L))

POST=(SRic restituisce col return la lista di nodoE ordinata che contiene tutti i nodi di vL)

NB: a) Le liste dei nodi di vL contenute nella lista di nodoE restituita da SRic devono essere tali che in ciascuna di esse i nodi siano tra loro nello stesso ordine che essi avevano in vL.

b) La funzione SRic deve usare la funzione insOrd dell'Esercizio 1 per costruire la lista di nodoE da restituire.

c) SRic e insOrd dovrebbero entrambe seguire la tipologia 1 delle funzioni ricorsive.

Correttezza (5 punti): scrivere la prova di correttezza della funzione indOrd.

Avvertenza generale:

Fare solo quello che viene chiesto esplicitamente nell'esercizio.

L'esercizio non richiede di creare nuovi nodi di tipo nodo. Quindi non fatelo. Richiede invece di creare nuovi nodi di tipo nodoE. Quindi fatelo.