

Esame di Programmazione del 20/9/2017

Il problema riguarda la eliminazione delle ripetizioni in una lista concatenata.

Esempio: Consideriamo la seguente lista $L=[0,1] \rightarrow [1,0] \rightarrow [2,0] \rightarrow [3,1] \rightarrow [4,1]$, in cui ogni nodo contiene 2 interi, il primo è il numero di sequenza del nodo nella lista, quindi 0,1,2, ecc., mentre il secondo è il consueto campo info del nodo. In L alcuni campi info dei nodi ripetono. Ci sono 3 nodi con campo info=1 (il primo, il penultimo e l'ultimo) e 2 nodi con campo info = 0 (il secondo e il terzo). **Vogliamo eliminare queste ripetizioni lasciando solo il primo nodo per ogni diverso valore di campo info.** Di L resterebbero quindi solo i seguenti 2 nodi: $[0,1] \rightarrow [1,0]$. Gli altri nodi non vanno deallocati, ma vanno organizzati in una lista Q di nodi di tipo nodoF (vedi programma dato) che contiene un valore FIFO capace di gestire una lista concatenata di nodi normali (di tipo nodo) attraverso i campi **primo** e **ultimo** che puntano al primo e all'ultimo nodo della lista. La lista Q del nostro esempio sarebbe costituita da 2 nodi, il primo che contiene un valore FIFO che gestisce la lista $[2,0]$ (quindi primo= $[2,0]$ e ultimo= $[2,0]$), mentre il secondo nodo contiene un valore FIFO che gestisce la lista $[3,1] \rightarrow [4,1]$ (con primo= $[3,1]$ e ultimo= $[4,1]$). E' importante osservare che:

- i) ciascun nodo di Q gestisce una lista di nodi con uguale campo info;
- ii) il primo nodo di Q gestisce i nodi con campo info=0 e il secondo nodo di Q quelli con campo info=1. Insomma le diverse liste gestite dai nodi di Q hanno info crescenti.

Consideriamo ora una lista L più lunga.

$L=[0,2] \rightarrow [1,0] \rightarrow [2,2] \rightarrow [3,0] \rightarrow [4,1] \rightarrow [5,2] \rightarrow [6,1] \rightarrow [7,2] \rightarrow [8,0] \rightarrow [9,1]$. Ci sono 3 diversi campi info in L: 2, 1 e 0. Tenendo solo il primo nodo per ciascun campo info, resta $L=[0,2] \rightarrow [1,0] \rightarrow [4,1]$, mentre Q ha 3 nodi nodoF, in cui il primo gestisce $[3,0] \rightarrow [8,0]$, il secondo gestisce $[6,1] \rightarrow [9,1]$ e il terzo gestisce $[2,2] \rightarrow [5,2] \rightarrow [7,2]$. Sottolineiamo nuovamente che i campi info crescono nelle liste gestite dai nodi di Q. Prima 0, poi 1 e infine 2.

Esercizio ricorsivo 1 (6) : si chiede di scrivere una funzione ricorsiva `nodoF* tieni_primo_ric(nodo*& L)` che soddisfa le seguenti specifiche:

PRE=(lista(L) è corretta, vL=L)

POST=(L contiene tanti nodi quanti sono i diversi campi info di vL e per ciascun campo info di vL contiene esattamente il primo nodo con quel valore di info)&&(la funzione restituisce col return una lista di nodi nodoF i cui nodi gestiscono i nodi ripetuti di vL nel modo spiegato nell'esempio)

Funzioni ricorsive ausiliarie (4+4): `tieni_primo_ric` deve usare le seguenti funzioni ricorsive ausiliarie:

- i) `FIFO eliminaR(nodo*&L, int x)` con PRE=(lista(L) corretta, vL=L) e POST=(L è vL da cui sono stati eliminati tutti i nodi con info=x)&&(viene restituito un valore FIFO che gestisce la lista dei nodi tolti nello stesso ordine che hanno in vL)
- ii) `nodoF* insR(nodoF*Q, FIFO x)` con PRE=(lista(Q) è corretta e ordinata rispetto ai campi info delle liste gestite dai suoi nodi, x gestisce lista non vuota) e POST=(restituisce Q a cui è stato aggiunto un nodoF che contiene x in modo da mantenere l'ordine dei campi info delle liste gestite dai suoi nodi)

Esercizio iterativo 1 (6): scrivere la funzione iterativa `nodoF* tieni_primo_iter(nodo*& L)` che soddisfa le PRE e POST date per `tieni_primo_ric`. La funzione può usare le funzioni ausiliarie ricorsive `eliminaR` e `insR`.

Esercizio iterativo 2 (5): scrivere la funzione iterativa `nodoF* insI(nodoF*Q, FIFO x)` che rispetta PRE e POST di `insR` del punto (ii) precedente.

Correttezza (2+3):

- 1) specificare l'invariante del ciclo principale di `tieni_primo_iter`;
- 2) delineare la prova induttiva della correttezza di `tieni_primo_ric`, assumendo che le funzioni ausiliarie che essa usa siano corrette rispetto alla PRE e POST date.