

Compito di Programmazione del 7 gennaio 2019 (Ultimo appello dell'anno 17-18)

Attenzione: se la vostra consegna NON passa i 4 test, allora SICURAMENTE ha qualcosa di sbagliato, se invece l'esame passa i test, NON DOVETE ASSOLUTAMENTE assumere che sia corretto. Quando qualcuno dei test fallisce, il moodle mostra l'input solo del primo test che fallisce.

Esercizio: viene data una lista concatenata L ed un array di interi non negativi A con dimA valori. Si vuole "affettare" L in dimA sottoliste di dimensioni pari ai valori in A. Quindi la prima "fetta" sarà costituita dai primi A[0] nodi di L, la seconda fetta dai successivi A[1] nodi di L e così via. E' possibile che A non basti ad affettare tutto L o al contrario che L non sia sufficientemente lunga per i valori di A. Quello che si deve fare in questi casi è descritto nell'esempio che segue.

Esempio: Sia L=4->3->10->7->2->5->6->8->9->1, dimA=5 e A=[2,2,1,0,3]. Allora le fette da tagliare sarebbero 5, le chiamiamo F1,F2,...F5, e il loro valore segue:

F1=4->3, F2=10->7, F3=2, F4 = vuota, F5= 2->5->6. Le fette vengono inserite in una lista concatenata LL di nodi di tipo nodoL (vedi programma dato). LL conterrà quindi 5 nodi, il primo punterà a F1, il secondo a F2, e così via.

Visto che A non affetta tutto L, alla fine dei conti vogliamo che L sia la parte restante di L. In questo esempio L deve diventare, 8->9->1.

Supponiamo ora che A sia [3,3,2,5,0] e che L sia come prima. In questo caso verrebbero fatte sono le 3 fette: F1=4->3->10, F2=7->2->5 e F3=6->8. La quarta fetta dovrebbe essere di lunghezza 5, ma in L restano solo 2 elementi e quindi la quarta fetta non viene tagliata e in L restano i 2 ultimi nodi della lista originale, e cioè, 9->1. In questo caso la quinta fetta non viene neppure considerata anche se è di lunghezza 0.

Per finire consideriamo il caso che A sia [3,3,4,0,1]. In questo caso verrebbero tagliate le 4 fette F1=4->3->10, F2=7->2->5 e F3=6->8->9->1 e F4= vuota. Alla fine L sarebbe vuota e il quinto elemento di A non avrebbe effetto.

Esame: L'esame richiede di risolvere l'esercizio appena descritto in modo ricorsivo e anche in modo iterativo. Le 2 funzioni da fare si devono chiamare affettaric e affettaiter. Entrambe devono obbedire al seguente prototipo e soddisfare le seguenti PRE e POST:

PRE=(lista(L) corretta, dimA >=0 e A contiene dimA interi non negativi (quindi è possibile anche il valore 0), vL= valore iniziale di L)

nodoL* affettaY(nodo*&L, int*A, int dimA) // Y = ric/iter

POST=(restituisce una lista di nodi di tipo nodoL che puntano alle fette di vL secondo quanto descritto nell'esempio precedente, L è quello che resta di vL una volta tolte le fette)

Consiglio: sia per l'esercizio ricorsivo che per quello iterativo conviene avere una funzione ausiliaria (rispettivamente, ric / iter) che aiuti a staccare le fette da L (quando possibile). E' possibile che servano anche altre funzioni ausiliarie che dovranno essere sempre ricorsive/iterative a seconda della parte dell'esame in cui vengono usate.

Correttezza:

- 1) Specificare l'invariante del ciclo principale di affettaiter
- 2) Descrivere la prova induttiva della correttezza di affettaric rispetto alle PRE e POST date.