

Esercizio simile al IV Compitino del 17/5/2016

Si tratta di leggere 3 interi, n_el , k e z , e poi di leggere n_el interi e contemporaneamente di costruire ricorsivamente una lista $L(Q)$ di n_el nodi che contengono nel loro campo info gli interi letti. Abbiamo visto come fare questa operazione, tramite la funzione F dell'esercizio 0 dell'11/5/2016. Ricordiamo che, se Q punta al primo nodo di una lista concatenata, con $L(Q)$ si denota l'intera lista.

Successivamente si vogliono rimuovere da $L(Q)$ tutti i nodi con $info=z$ meno gli ultimi k che devono invece rimanere in $L(Q)$. Questo va fatto solamente se $L(Q)$ contiene almeno k nodi con campo $info=z$. Altrimenti $L(Q)$ va lasciata com'è.

Esempi: se $n_el=10$, $k=3$ e $z=0$, e la lista $L(Q)$ di 10 nodi fosse: $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 0 \rightarrow 0 \rightarrow 1 \rightarrow 0 \rightarrow 1 \rightarrow 1$, allora la funzione da fare dovrebbe restituire col return la lista che contiene il primo nodo con $info=0$ e la lista $L(Q)$ diventerebbe: $1 \rightarrow 2 \rightarrow 3 \rightarrow 0 \rightarrow 0 \rightarrow 1 \rightarrow 0 \rightarrow 1 \rightarrow 1$. Cioè gli ultimi 3 nodi con $info=0$ restano in $L(Q)$ e viene rimosso da $L(Q)$ solo il primo nodo con $info=0$ che viene restituito col return.

Se $n_el=10$, $k=4$ e $z=0$ e la lista $L(Q)$ fosse: $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 0 \rightarrow 0 \rightarrow 1 \rightarrow 0 \rightarrow 1 \rightarrow 1$, allora visto che in $L(Q)$ ci sono esattamente 4 nodi con $info=0$, allora la lista $L(Q)$ non cambia e viene restituito 0 col return.

Infine, se $n_el=10$, $k=0$ e $z=0$ e la lista $L(Q)$ fosse: $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 0 \rightarrow 0 \rightarrow 1 \rightarrow 0 \rightarrow 1 \rightarrow 1$, allora tutti i nodi con $info=0$ verrebbero rimossi da $L(Q)$ e restituiti col return. In sostanza col return andrebbe restituita la lista $0 \rightarrow 0 \rightarrow 0 \rightarrow 0$ e $L(Q)$ diventerebbe: $1 \rightarrow 2 \rightarrow 3 \rightarrow 1 \rightarrow 1 \rightarrow 1$.

La funzione che esegue queste operazioni è **ricorsiva** e deve essere conforme alla seguente specifica:

PRE=($L(Q)$ è lista corretta, $k \geq 0$, z def., ck def., $vL(Q)=L(Q)$, $vk=k$ e $vck=ck$)

nodo* IUK(nodo*&Q, int k, int z, int & ck)// IUK sta per lascia ultimi k (nodi)

POST=(se $vL(Q)$ contiene x nodi con $info=z$ e $vck+x \geq k$, allora ci sono 2 casi:

a) se $x-k > 0$ allora $ck=0$ e viene restituita con return la lista dei primi $x-k$ nodi di $vL(Q)$ con $info=z$ e $L(Q)$ è quello che resta di $vL(Q)$ dopo aver tolto questi nodi.

b) se $x-k \leq 0$ allora $ck=k-x$ e col return viene restituito 0, mentre $L(Q)=vL(Q)$) &&

(se $vL(Q)$ contiene x nodi con $info=z$ e $vck+x < k$, allora $ck=vck$, $L(Q)=vL(Q)$ e viene restituito 0 col return)

Importante: Si osservi con attenzione la POST. Infatti essa indica come usare il parametro ck (passato per riferimento). All'andata ck conta i nodi con $info=z$ in modo da sapere se ce ne sono almeno k . Al ritorno invece indica quanti nodi con $info=z$ devono ancora venire lasciati al loro posto percorrendo la lista $L(Q)$ a ritroso. Quindi $k-ck$ è il numero degli ultimi nodi di $vL(Q)$ con $info=0$ che sono stati lasciati al loro posto. Quindi, se $ck=0$ significa che sono stati lasciati al loro posto gli ultimi k nodi con $info=z$ e perciò andranno tolti dalla lista tutti i nodi con $info=z$ che verranno incontrati durante il ritorno della ricorsione.

Attenzione: viene fortemente sconsigliato l'uso di una funzione che conta quanti nodi con $info=z$ sono presenti in $L(Q)$. Il programma non deve creare o distruggere nodi. Ma usare solo quelli inizialmente in $vL(Q)$.

Correttezza: scrivere la prova di correttezza induttiva di IUK rispetto a PRE e POST.