

III Compitino di Programmazione 30/5/2018

L'esercizio richiede di traversare in ordine infisso un albero binario costruendo una lista concatenata i cui nodi puntano a nodi dell'albero che sono attraversati, seguendo il seguente criterio: viene puntato un nodo ogni k nodi attraversati, per un qualche $k > 0$.

Esempio. Sia questo l'albero: $1(2(4(8(_, _), _), 6(0(_, _), _)), 3(5(9(_, _), _), 7(_, _)))$, allora i nodi ordinati secondo l'ordine infisso sono: 8 4 2 0 6 1 9 5 3 7. Se $k=2$, allora la lista da costruire deve avere 5 nodi che puntano, da sinistra a destra ai nodi, 4, 0, 1, 5, e 7 dell'albero. Se $k=1$ allora avremmo una lista lunga 10 che punta a tutti i nodi dell'albero, mentre se $k=3$, avremmo una lista di 3 nodi che puntano ai nodi, 2, 1, e 3 dell'albero.

Visto che si tratta contemporaneamente una lista concatenata ed un albero, nel programma dato vengono dichiarate 2 strutture: `nodo` (per la lista) e `nodoA` (per l'albero). Osservate che il campo `info` di `nodo` ha tipo `nodoA*` per puntare ai nodi dell'albero.

La funzione ricorsiva che deve produrre la lista desiderata deve obbedire alle seguenti specifiche:

`PRE=(albero(r) è ben formato, $k > 0$, $1 \leq n \leq k$)`

`nodo* B(nodoA*r, int k, int& n)`

`POST=(restituisce la lista concatenata i cui nodi puntano ai nodi dell'albero ordinati secondo l'ordine infisso, e tale che il primo nodo della lista punti al nodo n -esimo di albero(r) secondo l'ordine infisso, e i successivi nodi puntano ad un nodo di albero(r) ogni k nodi sempre secondo l'ordine infisso) && (n , $1 \leq n \leq k$, indica che si devono attraversare $n-1$ nodi prima di arrivare a quello che va puntato dal prossimo nodo della lista)`

Esempio. Se usiamo lo stesso albero dell'esempio precedente, i cui nodi in ordine infisso sono, 8 4 2 0 6 1 9 5 3 7, e se consideriamo un'invocazione di `B` sulla radice dell'albero, con $k=3$ e $n=2$, allora la lista restituita da `B` dovrebbe puntare ai nodi 4, 6, e 5 e alla fine, $n=1$.

Consiglio: serve una funzione capace di concatenare 2 liste date. Va fatta ricorsiva.

Correttezza: fare la prova induttiva della correttezza di `B`.