

Esame di Programmazione del 17/9/2019 Parte iterativa

L'esercizio riguarda l'attraversamento di un albero binario in larghezza. Si parte del livello 0 che contiene solo la radice dell'albero, poi il livello 1 che contiene i figli della radice, il livello 2 che contiene i figli dei nodi del livello 1 e così via. Ogni livello è attraversato da **destra a sinistra**.

Per eseguire in modo iterativo l'attraversamento di un albero in larghezza, conviene usare una lista concatenata i cui nodi puntino ai nodi dell'albero. Inizialmente la lista contiene un puntatore alla radice dell'albero. Ad ogni iterazione si deve estrarre dalla lista il primo nodo e aggiungere in fondo alla lista i puntatori ai suoi figli (quelli che ci sono) nell'ordine che garantisca di percorrere i livelli da destra a sinistra. Viste le operazioni da fare, conviene gestire la lista con una coda. La struct coda, le funzioni di pop e push_end sulla coda e la struct nodoEx dei puntatori ai nodi che sono le componenti della lista gestita dalla coda, sono tutti nel programma dato. Si osservi che la struttura nodoEx contiene anche un campo liv di tipo int che serve a contenere il livello del nodo puntato. Si osservi anche che i figli di un nodo di livello k hanno ovviamente livello k+1.

Esercizio da fare: si chiede di realizzare la funzione **iterativa** scanliv (scandisci i livelli) che riceve un albero R e percorre R in larghezza percorrendo ciascun livello **da destra a sinistra**. Deve anche restituire il numero di livelli presenti in R e deve inserire valori negli array X e T come segue:

--per ogni livello i deve restituire in X[i] il numero di nodi presenti in quel livello

--T deve contenere il campo info dei nodi di R nell'ordine in cui sono percorsi (in larghezza da destra a sinistra)

Esempio 1: per l'albero R=[10]([5](_,_),[20](_,[30]([21](_,_),[31](_,[32](_,_)))))

Il numero di livelli è 5, X=[1,2,1,2,1] che indica che c'è 1 nodo al livello 0 (X[0]), 2 al livello 1 (X[1]), 1 nodo al livello 2, 2 al 3 e 1 al 4. Inoltre T=[10,20,5,30,31,21,32]. Insomma i campi info dei nodi di R nell'ordine per livelli e con ogni livello percorso da destra a sinistra.

La funzione iterativa scanliv deve soddisfare le seguenti specifiche:

PRE=(albero(R) è un albero binario ben formato, X e T contengono un numero di posizioni pari al numero dei nodi di albero(R))

void scanliv(nodo*R, int& nliv, int*X, int*T,)

POST=(nliv è il numero di livelli di albero(R) e X e T contengono i valori descritti nell'Esempio 1, ovviamente per l'albero(R) in input)

Correttezza:

- i) scrivere un invariante significativo per il ciclo principale di scanliv
- ii) scrivere la correttezza del ciclo principale di scanliv che usi l'invariante di (i)