

## Esercizio 2 del 29/5/2017

Si tratta di leggere  $n > 0$  numeri da cin e costruire un albero con  $n$  nodi che contengano gli  $n$  valori letti nel loro campo info. Partendo dall'albero vuoto (nodo\*  $r=0$ ) si dovrà inserire un nuovo nodo scegliendo sempre di inserirlo come radice, se l'albero è vuoto, oppure a sinistra o a destra della radice a seconda che nel sottoalbero sinistro o destro della radice ci siano meno nodi. Questa procedura va applicata ricorsivamente. Vediamo un esempio:

**Esempio 1.** Sia  $r=0$  e assumiamo di leggere 3 da cin, quindi  $\text{albero}(r)$  deve diventare  $3(\_,\_)$  a indicare che c'è solo la radice con  $\text{info}=3$  e  $\text{left}$  e  $\text{right}=0$ . Poi leggiamo 10, il sottoalbero sinistro e destro della radice hanno entrambi 0 nodi, quindi mettiamo il nodo con 10 nel sottoalbero sinistro. Per cui  $\text{albero}(r) = 3(10(\_,\_),\_)$ . Poi leggiamo 12 e  $\text{albero}(r)=3(10(\_,\_),12(\_,\_))$ . Di seguito leggiamo 2 e  $\text{albero}(r)=3(10(2(\_,\_),\_),12(\_,\_))$ . Infine se leggiamo 3 e 4 avremo prima  $3(10(2(\_,\_),\_),12(3(\_,\_),\_))$  e poi  $3(10(2(\_,\_),4(\_,\_)),12(3(\_,\_),\_))$ . Si osservi che procedendo in questo modo, l'albero crescerà in modo bilanciato nel senso che, per ogni nodo interno  $m$ , il numero di nodi del sottoalbero sinistro e destro di  $m$  avranno al massimo differenza 1.

Si richiede di scrivere la funzione ricorsiva  $\text{nodo* alberobil}(\text{nodo* } r, \text{int } k)$  che soddisfa le seguenti pre e postcondizioni:

PRE= $(\text{albero}(r)$  è corretto e bilanciato nel senso spiegato nell'Esempio 1,  $\text{valbero}(r)=\text{albero}(r)$ )

POST= $(\text{albero}(r)$  è corretto e ancora bilanciato ed è ottenuto da  $\text{valbero}(r)$  aggiungendogli una nuova foglia con  $\text{info}=k$ )

Il programma deve invocare  $\text{alberobil}$   $n$  volte, per inserire gli  $n$  nodi con gli  $n$  valori letti da cin. Per farlo è necessario scrivere una funzione  $\text{buildtree}$  con le seguenti specifiche:

PRE= $(\text{albero}(r)$  è corretto e bilanciato,  $n \geq 0$ ,  $\text{valbero}(r)=\text{albero}(r)$ )

Nodo\*  $\text{buildtree}(\text{nodo* } r, \text{int } n)$

POST= $(\text{restituisce valbero}(r)$  con  $n$  nodi aggiuntivi inseriti in modo da conservare il bilanciamento)

L'albero finale prodotto da  $\text{buildtree}$  deve venire stampato in modo lineare (come mostrato nell'Esempio 1) invocando la funzione  $\text{stampalin}(\text{nodo* } r)$ , vista in classe. E' molto utile usare almeno una funzione ricorsiva ausiliaria. Nel caso lo facciate, dovete fornire la sua PRE e POST.

**Correttezza:** dimostrare la correttezza della funzione  $\text{alberobil}$  rispetto alla PRE e POST date.