

Corso di laurea in Matematica

Insegnamento di Informatica generale Canale Lb – Z

Docenti: G. Bongiovanni, S. Silvestri

Prova di laboratorio del 19 luglio 2012

ESERCIZIO

Si vuole gestire un insieme dinamico attraverso un albero binario di ricerca. Tale albero può contenere chiavi ripetute. Per evitare che vi siano più elementi con la stessa chiave, ogni elemento contiene un campo aggiuntivo *mul* che rappresenta il numero di elementi con quella chiave che sono stati inseriti. Ad esempio, quindi, un elemento dell'albero con chiave 3 e campo *mul* pari a 5 indica che l'albero contiene 5 elementi con chiave 3.

La struttura dati utilizzata è la seguente:

```
typedef struct nodo_mul{
    int info;
    int mul;
    struct struct nodo_mul *sx, *dx;
} nodo_mul;
```

Si devono definire le seguenti funzioni:

nodo_mul * insert(nodo_mul * t, int x)

Tale funzione, presa in input la radice dell'albero *t* ed il valore della chiave *x* dell'elemento da inserire, ritorna la radice dell'albero con l'elemento inserito. Se non esiste in *t* un elemento con chiave *x*, verrà creato un nuovo elemento ed inserito nell'albero (seguendo le regole di inserimento di un ABR), se invece esiste già un elemento con chiave *x*, la funzione dovrà solo incrementare il campo *mul* di tale elemento.

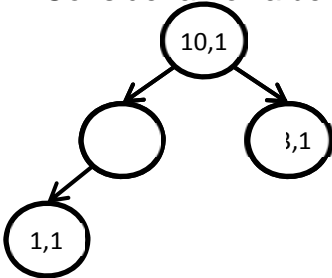
void * inorder(nodo_mul * t)

Tale funzione esegue la visita in order dell'albero. Ogni elemento deve essere stampato un numero di volte pari a quelle del suo campo *mul*.

Esempio

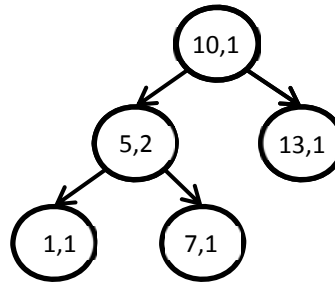
Indichiamo con (x,y) è un elemento con chiave x e campo mul pari a y .

1. Consideriamo l'albero:



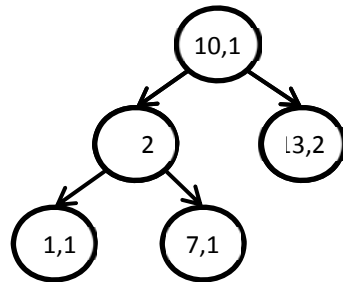
La stampa in order è: 1 5 5 10 13

2. Chiamando $t = \text{insert}(t, 7)$, si ottiene:



La stampa in order è: 1 5 5 7 10 13

3. Chiamando $t = \text{insert}(t, 13)$, si ottiene:



La stampa in order è: 1 5 5 7 10 13 13