

PROGRAMMAZIONE II– CANALE P-Z  
**Esame del 12/07/02**

La prova dura 1h.

Supponiamo di avere definito i tipi

```
struct nodo {
    int val;
    struct nodo *sn, *dx;
};
typedef struct nodo NODO;
typedef NODO *ALBBIN;
```

per rappresentare alberi binari.

**Esercizio 1.** Scrivere la funzione

```
ALBBIN cerca(ALBBIN t, int i);
```

che ritorna il puntatore al nodo dell'albero contenente  $i$  come etichetta, se un tale nodo esiste, o `NULL` altrimenti.

**Esercizio 2.** Supponiamo ora che gli alberi binari che considereremo abbiano le seguenti caratteristiche: per ogni nodo dell'albero  $p$  dell'albero, se  $v$  è il valore dell'etichetta di  $p$ , allora tutti i nodi del sottoalbero sinistro di  $p$  hanno valori minori di  $v$ , mentre tutti i nodi del sottoalbero destro di  $p$  hanno valori maggiori di  $v$ .

Scrivere la funzione

```
ALBBIN cerca2(ALBBIN t, int i);
```

analogamente alla `cerca` del precedente esercizio, ma per gli alberi con le suddette caratteristiche.

**Esercizio 3.** Scrivere la funzione

```
void stampaord(ALBBIN t);
```

che prende un albero con le caratteristiche descritte nel precedente esercizio e stampa la lista ordinata delle sue etichette.

**Esercizio 4.** Supponiamo che l'albero contenga  $n$  elementi ed abbia profondità massima  $d$ .

1. Prendiamo in esame la `cerca`. Qual è il numero massimo di etichette dell'albero con cui occorre confrontare  $i$  prima di trovare il nodo che contiene  $i$  o concludere che tale nodo non esiste?
2. E nel caso della `cerca2`, qual è il numero massimo di confronti?
3. C'è differenza nel caso peggiore?
4. Cambia nulla se considero alberi bilanciati?