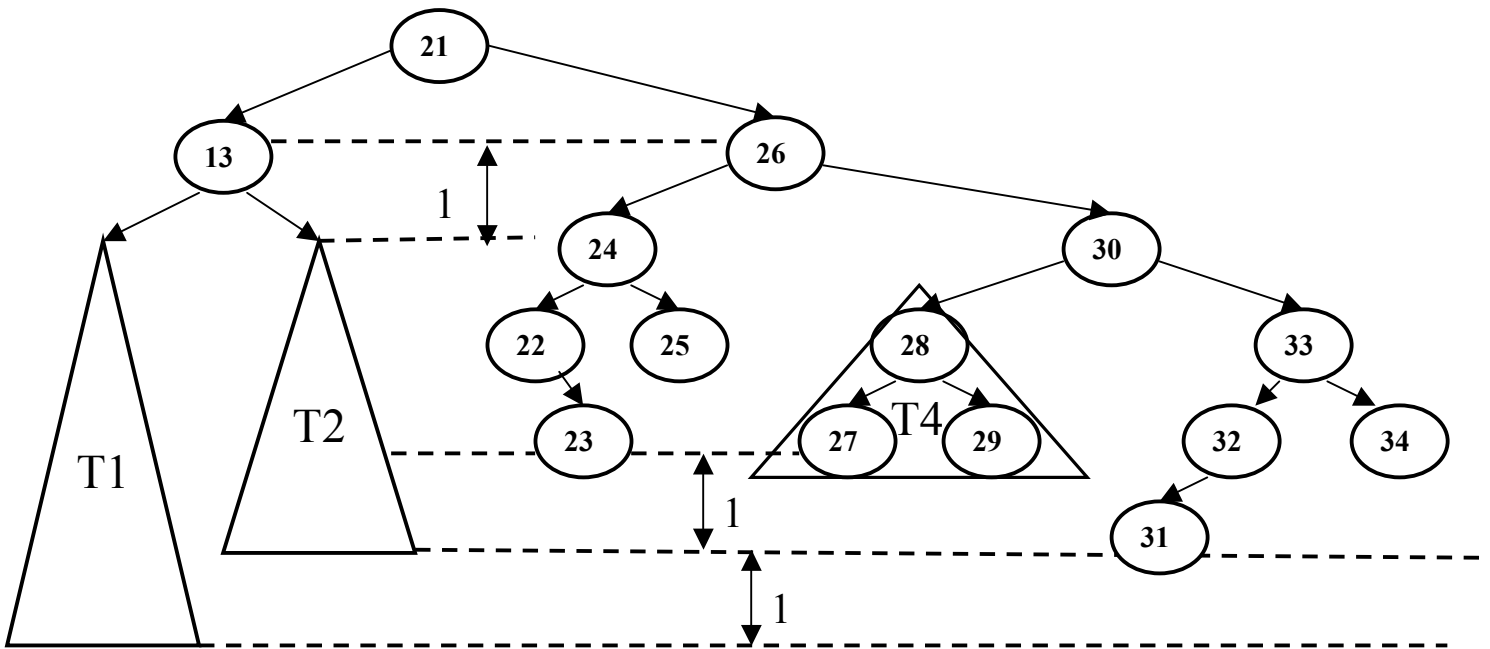


ALGORITMI E LABORATORIO – prova scritta del 17 luglio 2008 – Parte 1

Cognome e Nome _____ Matr. _____

1. (~ punti 7) Nell'albero AVL sottostante si elimini l'elemento 21, disegnando l'albero dopo ognuna delle rotazioni effettuate.



NOTA BENE: Per semplificare i disegni, potete racchiudere in triangoli, come nella figura, i sottoalberi che rimangono invariati, assegnargli un nome (T3, T4, ecc.), e poi disegnare semplicemente i triangoli coi nomi, senza riportare i nodi interni ad essi.

2 (~punti 6). Si definisca il metodo `public ElemType add(ElemType el)` della classe `BinSearchTree`: il metodo deve realizzare una versione iterativa dell'inserimento in un albero binario di ricerca in cui le foglie nulle sono rappresentate dalla costante `null` (e non da un nodo fittizio); se un elemento con la stessa chiave è già presente nell'albero, lo sostituisce con il nuovo elemento e restituisce il vecchio elemento; altrimenti inserisce il nuovo elemento e restituisce `null`.

3 (~ punti 4). Si consideri il seguente algoritmo divide-et-impera:

```
static void divimp(int n) {
    if(n <= 1) return;
    else {
        int count = 0;
        for(int i = 1; i <= 8; i++) divimp(n/2);
        int ncubo = n*n*n;
        for(int i = 1; i <= ncubo; i++) count++;
    }
}
```

Si scrivano le equazioni di ricorrenza per il tempo di calcolo e, applicando il teorema master, si ricavi la complessità temporale dell'algoritmo.

4. (~ punti 4). Si spieghi in che cosa consiste la tecnica del raddoppiamento/dimezzamento negli array dinamici, e si dimostri che adottando tale tecnica la complessità ammortizzata delle operazioni di inserimento e cancellazione risulta costante. In particolare, si spieghi perché – come scritto sulle slides – "il costo h della copiatura di h elementi per la riallocazione si ripartisce su almeno $h/2$ precedenti inserimenti senza riallocazione".