# Alberi

## Grado di sbilanciamento

Dato un albero binario $T$, il grado di sbilanciamento di un nodo $v$ è pari alla differenza, in valore assoluto, fra il numero di foglie presenti nel sottoalbero sinistro di $v$ e il numero di foglie presenti nel sottoalbero destro di $v$. Il grado di sbilanciamento dell'albero $T$ è pari al massimo grado di sbilanciamento dei nodi di $T$.

Scrivere un algoritmo che dato un albero $T$ restituisce il grado di sbilanciamento dell'albero. Discuterne correttezza e complessità.

# Alberi

## Albero livello-valore

Scrivere un algoritmo che preso in input un albero binario $T$ i cui nodi sono associati ad un valore intero $T.key$, restituisca il numero di nodi dell'albero il cui valore è pari al livello del nodo. Vi ricordo che il livello del nodo è pari al numero di archi che devono essere attraversati per raggiungere il nodo dalla radice. Per cui la radice ha livello 0, i suoi figli hanno livello 1, etc.

# Esercizi

## Sequenza specchiata

Dato un vettore $V$ di $n$ interi appartenenti all'insieme $\{0, 1\}$, si scriva un algoritmo che restituisca la lunghezza del più lungo sottovettore contiguo formato da $k$ valori 0 seguiti da $k$ valori 1.

Si noti che è possibile che esistano altri valori 0 prima del sottovettore così individuato, oppure altri valori 1 dopo il sottovettore, ma non è possibile che si verifichino entrambe le estensioni, altrimenti il sottovettore non sarebbe massimale. Discutere correttezza e complessità dell'algoritmo proposto.

Esempio: per l'input `0011111110001110000`, l'algoritmo deve restituire 6.

# Analisi – Ricorrenze

## Algoritmo di selezione deterministico

Si consideri la seguente equazione di ricorrenza:

$$T(n) = \begin{cases} 11/5n + T(\lfloor n/5 \rfloor) + T(\lfloor 7n/10 \rfloor) & n > 1 \\ 1 & n \leq 1 \end{cases}$$

Individuare limiti inferiori e superiori tramite il metodo di sostituzione.

# Spoiler alert!

# Alberi - Grado di sbilanciamento

**integer**, **integer** unbalance(TREE $T$)

**if** $T =$ **nil then**
  **return** $(0, 0)$

**if** $T.left =$ **nil and** $T.right =$ **nil then**
  **return** $(1, 0)$

$L_{leafs}, L_{max} \leftarrow \mathsf{unbalance}(T.left)$
$R_{leafs}, R_{max} \leftarrow \mathsf{unbalance}(T.right)$
**return** $(L_{leafs} + R_{leafs}, \mathsf{max}(L_{max}, R_{max}, |L_{leafs} - R_{leafs}|))$

## Albero livello-valore

Una semplice visita posticipata risolve il problema. La complessità è ovviamente $O(n)$.

```
sameLevel(Tree T, integer ℓ)
integer
if T = nil then
    return 0
else
    integer count ← sameLevel(T.right(), ℓ+1) + sameLevel(T.left(), ℓ+1)
    if T.key = ℓ then
        count ← count + 1
    return count
```

# Sequenza specchiata

Non sono necessarie particolari tecniche per risolvere questo esercizio; è sufficiente fare una scansione di costo lineare $\Theta(n)$.

```
integer mirror(integer [] V, integer n)
count0, count1 ← 0
largest ← 0
for i ← 1 to n do
    if V[i] = 0 then
        if count1 > 0 then
            count0 ← 0
            count1 ← 0
        count0 ← count0 + 1
    else
        count1 ← count1 + 1
        largest ← max(largest, min(count0, count1))
return largest
```

# Ricorrenza $T(n) = 1/5n + T(\lfloor n/5 \rfloor) + T(\lfloor 7n/10 \rfloor)$

Caso base

$$
\begin{aligned}
T(0) &= 1 \leq c \cdot 0 \Leftrightarrow 1 \leq 0 \qquad \text{Falso!} \\
T(1) &= 1 \leq c \cdot 1 \Leftrightarrow c \geq 1 \\
T(2) &= 22/5 + T(\lfloor 2/5 \rfloor) + T(\lfloor 14/10 \rfloor) \\
&= 22/5 + T(0) + T(1) = 32/5 \leq c \cdot 2 \Leftrightarrow c \geq 32/10 \\
T(3) &= 33/5 + T(\lfloor 3/5 \rfloor) + T(\lfloor 21/10 \rfloor) \\
&= 33/5 + T(0) + T(2) = 70/5 \leq c \cdot 3 \Leftrightarrow c \geq 70/15 \\
T(4) &= 44/5 + T(\lfloor 4/5 \rfloor) + T(\lfloor 28/10 \rfloor) \\
&= 44/5 + T(0) + T(2) = 81/5 \leq c \cdot 4 \Leftrightarrow c \geq 81/20 \\
T(5) &= 55/5 + T(\lfloor 5/5 \rfloor) + T(\lfloor 35/10 \rfloor) = 55/5 + T(1) + T(3)
\end{aligned}
$$

# Ricorrenza $T(n) = 1/5n + T(\lfloor n/5 \rfloor) + T(\lfloor 7n/10 \rfloor)$

Passo ricorsivo

$$\begin{aligned} T(n) &\leq 11/5n + c\lfloor n/5 \rfloor + c\lfloor 7n/10 \rfloor \\ &\leq 11/5n + 1/5cn + 7/10cn \\ &= 9/10cn + 22/10n \leq cn \end{aligned}$$

L'ultima disequazione è vera per $c \geq 22$.