# Alberi – Albero livello–valore

Scrivere un algoritmo che preso in input un albero binario $T$ i cui nodi sono associati ad un valore intero $T.value$, restituisca il numero di nodi dell'albero il cui valore è uguale 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.

# Alberi – Cammino radice–discendente crescente

Dato un albero binario contenente interi, scrivere un algoritmo che restituisca la lunghezza del più lungo cammino monotono crescente radice-discendente, dove:

- il discendente non è necessariamente foglia;
- con lunghezza si intende il numero totale di archi attraversati;
- con monotona crescente si intende che i valori contenuti nei nodi della sequenza devono essere ordinati in senso crescente da radice a discendente.

Discuterne correttezza e complessità.

# Alberi – Grado di sbilanciamento

Si consideri 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 quelle 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 $T$, restituisca il grado di sbilanciamento dell'albero. Discuterne correttezza e complessità.

Nota: In pseudocodice, è possibile restituire una coppia di valori:

**(int, int)** fun1(TREE T)

```
[...]
return (a, b)
```

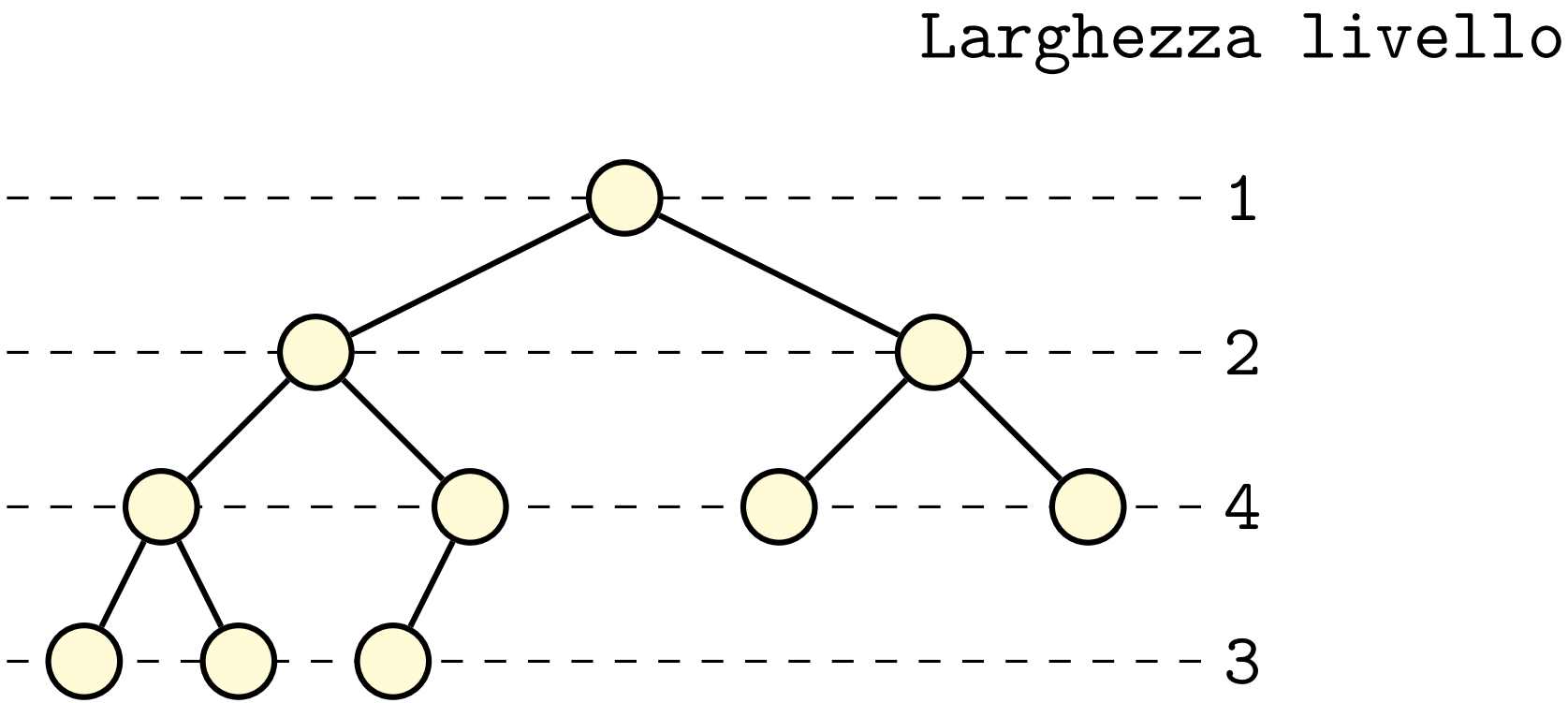
**int** fun2(TREE $T$)

```
int, int x, y = fun1(TREE T)
return y
```

# Alberi – Larghezza albero

La larghezza di un albero $T$ è il numero massimo di nodi di $T$ che stanno tutti al medesimo livello.

Scrivere un algoritmo che restituisca la larghezza di un albero ordinato $T$ contenente $n$ nodi.



# SortinoSort

Il professor Sortino ha inventato un nuovo algoritmo di ordinamento.

- Il vettore di input viene diviso in tre parti, di dimensioni circa $n/3$.
- Vengono ordinati ricorsivamente i primi due terzi, i secondi due terzi, e infine di nuovo i primi due terzi.

---

SortinoSort(**int**[ ] $A$, **int** $i$, **int** $j$)

---

**if** $j - i + 1 \leq 6$ **then**
    InsertionSort$(A, i, j)$
**else**
    **int** $s = \lceil (j - i + 1)/3 \rceil$
    SortinoSort$(A, i, i + 2s - 1)$
    SortinoSort$(A, i + s, j)$
    SortinoSort$(A, i, i + 2s - 1)$

---

# SortinoSort

1. Qual è la complessità di questo algoritmo? Il Prof. Sortino finirà nella prossima edizione del mio libro?
2. (Difficile, Opzionale) Dimostrare per induzione che questo algoritmo è corretto. Per comodità, assumete pure che tutti i valori siano distinti.