Prof. L. Margara **Scritto del 18/06/2019**

Algoritmi e Strutture Dati

Esercizio 1.[11 punti]
Calcolare la formula esatta per $T(n)$ definita come segue:

$$T(n) = \begin{cases} 1 & \text{if } n = 1, \\ T(n-1) + 7^n & \text{if } n > 1. \end{cases}$$

Dimostrare formalmente (per induzione) che il risultato trovato è corretto.

Esercizio 2.[11 punti]
Data una sequenza di n numeri interi maggiori di zero $x_1, ..., x_n$ diciamo che (x_i, x_{i+1}, x_{i+2}) è una terna di tipo A se $x_{i+1} = x_i + x_{i+2}$. Scrivere una funzione **ricorsiva** in **pseudo-codice** che utilizzi la **tecnica divide-et-impera** e che calcoli quante terne di tipo A sono contenute in una sequenza di n numeri interi $x_1, ..., x_n$ ricevuta in input. Valutare in ordine di grandezza il costo computazionale dell'algoritmo proposto.

Esercizio 3.[11 punti]
Scrivere in pseudo codice una FUNZIONE RICORSIVA che prenda in input il puntatore alla radice di un albero binario e restituisca l'intero positivo n definito come segue.
n è la lunghezza della più lunga sequenza *omogenea verticale* di nodi dell'albero.
Una sequenza di nodi $u_1, u_2, \ldots, u_n$ è verticale se u_1 è la radice dell'albero e u_i è il padre di u_{i+1} con $1 \leq i \leq n-1$.
Una sequenza di nodi $u_1, u_2, \ldots, u_n$ è omogenea se il valore di u_i è uguale al valore di $u_{i+1} - 1$ con $1 \leq i \leq n-1$.
(Si noti che se l'albero è vuoto il risultato è 0 e che se l'albero ha un solo nodo il risultato è 1)
Argomentare la sua correttezza e analizzare il suo costo computazionale. Vietato usare variabili globali !!! Usare un solo parametro !!!

Traccia della soluzione dell'esercizio 1.
$T(n) = \frac{7^{n+1}-43}{6}$
Dimostrazione per induzione.

Traccia Soluzione esercizio 2.

```
CONTA-TERNE(A, i, j)
if j - i < 2
    return 0
else
    p = posizione centrale tra i e j
    t1 = CONTA-TERNE(A, i, p)
    t2 = CONTA-TERNE(A, p + 1, j)
    t3 = numero di terne a cavallo della posizione p
    return t1 + t2 + t3
```

Equazione Ricorsiva:

$$T(n) = \begin{cases} c_1 & \text{if } n \leq c_2, \\ T(n/2) + T(n/2) + c_3 & \text{if } n > c_2. \end{cases}$$

Soluzione: $T(n) = \Theta(n)$

Traccia Soluzione esercizio.

```
LUNGHEZZA-SEQUENZA(p)
if p == NIL
    return 0
else
    l = r = 0
    if p.left ≠ NIL and p.left.key == p.key + 1
        l = LUNGHEZZA-SEQUENZA(p.left)
    if p.right ≠ NIL and p.right.key == p.key + 1
        r = LUNGHEZZA-SEQUENZA(p.right)
    return 1 + max(l, r)
```

n = numero di nodi dell'albero
Costo computazionale: $T(n) = \Theta(n)$