

1. Sea n un número natural y considere el siguiente código en Python que define la función recursiva $F(n)$:

```
Def F(n):
    if n == 0:
        return 1
    else:
        return 2n + 1 + F(n-1)
```

1.5

a) [0.5pts.] Escriba el paso a paso de $F(3)$.

b) [1pt.] Demuestre que $F(n) = (n + 1)^2$.

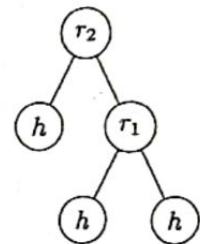
2. Sea A un árbol binario y asuma la definición de las siguientes funciones:

- Num_Nodos : Número de nodos de A .
- Num_Hojas : Número de hojas de A .

a) [0.5pts.] Escriba el paso a paso de $Num_Hojas(r_2)$

b) [1pt.] Demuestre por inducción estructural que:

$$Num_Nodos(A) = 2 * Num_Hojas(A) - 1$$



3. [1pt.] Escriba el paso a paso de

$$\text{Sust}[\neg(p \rightarrow q), p, \neg r]$$

4. [1pt.] Sea A una fórmula representada como un árbol y asuma la definición de las siguientes funciones:

- $C(A)$: Número de ocurrencias de conectivos (tanto unarios como binarios) en A .
- $Atom(A)$: Número de ocurrencias de letras proposicionales en A .

Demuestre por inducción estructural que:

$$Atom(A) \leq 2^{C(A)}$$

[Ayuda: Si $x, y \geq 1$, observe que $2^x \leq 2^{x+1}$ y que $2^x + 2^y \leq 2^x * 2^y \leq 2^{x+y+1}$.]