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

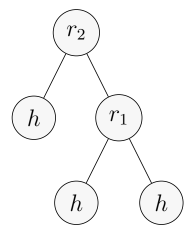
función F(n):
  Si $n == 0$
    retornar 1
  Si no
    retornar 2^n + F($n-1$)

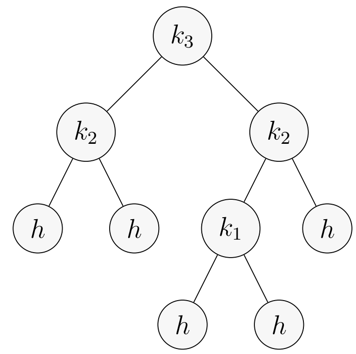
a) Escriba el paso a paso de $F(3)$.

b) Demuestre que $F(n) = 2^{n+1} - 1$

EJERCICIO 2: Sea A un árbol binario.

a) Defina de manera recursiva la función $Corta[A]$, la cual cuenta el número de aristas de la rama más corta de A. Observe que, por ejemplo, $Corta[r_2] = 1$ y $Corta[k_3] = 2$.





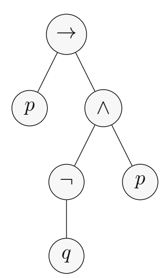
b) Presente el paso a paso de esta función sobre el árbol r_2.

c) Asuma la definición recursiva de la función $Num\_Aristas(A)$, la cual cuenta el número total de aristas de A. Demuestre por inducción estructural que

$$Corta(A) \leq \frac{Num\_Aristas(A)}{2}$$

EJERCICIO 3: Sea `I={'p':True, 'q':False}`. Escriba el paso a paso de $A_4.valor(I)$ donde:

$$A_0 = \text{TREE}(p, \text{NULL}, \text{NULL})$$
$$A_1 = \text{TREE}(q, \text{NULL}, \text{NULL})$$
$$A_2 = \text{TREE}(\neg, \text{NULL}, A_1)$$
$$A_3 = \text{TREE}(\wedge, A_2, A_0)$$
$$A_4 = \text{TREE}(\rightarrow, A_0, A_3)$$



EJERCICIO 4: Sea A una fórmula representada como un árbol y asuma la definición de las siguientes funciones:

- $num\_bin(A)$: Número de ocurrencias de conectivos binarios en A.
- $num\_negs(A)$: Número de ocurrencias de negaciones en A.
- $num\_letras(A)$: Número de ocurrencias de letras proposicionales en A.
- $str(A)$: Cadena que representa la notación inorder de A.
- $len(c)$: Cantidad de símbolos en la cadena c.

Demuestre por inducción estructural que:

$$len(str(A)) = 3 * A.num\_bin() + A.num\_negs() + A.num\_letras()$$