Universidad del Norte

IST 4330 - ESTRUCTURAS DISCRETAS

27 de agosto de 2020

Opciones para trabajar con Python

- **Descargar Anaconda:**
  Link: https://www.anaconda.com/products/individual Este se puede usar en su propio equipo. Es software installable (y open source).
- **Google Colab**
  Link: https://colab.research.google.com/ Este es de la suit de Google, se usa en la nube. Y nos da una capacidad de almacenamiento bastante generosa.
- **Repl.it**
- **PyCharm**

Ambos (Colab y Anaconda) trabajan con Jupyter, por lo que se puede usar un mismo *notebook* en ambos.

Relaciones de Recurrencia

Dada la secuencia

$$\langle 1,\, 3,\, 7,\, 15,\, 31,\, 63,\, 127, \ldots \rangle$$

$$f_0 = 1,\; f_1 = 3,\; \ldots$$

definir algunas relaciones de recurrencia.

Nos damos cuenta de que una posible fórmula es "el doble del anterior más uno". Es decir:

$$f_n = 2f_{n-1} + 1,\; n > 0,\; f_0 = 1.$$

También es posible definir otras...

$$f_n = 2f_{n-2} + f_{n-1} + 2,$$

$1 + 3 + 7 + \underbrace{4}_{g_n} = 15$

$3 + 7 + 15 + \underbrace{6}_{g_n} = 31$

$7 + 15 + 31 + \underbrace{10}_{g_n} = 63$

$15 + 31 + 63 + \underbrace{18}_{g_n} = 127$

$31 + 63 + 127 + \underbrace{34}_{g_n} = 255$

Inicialmente, tenemos:

$$f_n = f_{n-1} + f_{n-2} + f_{n-3} + g(n)$$

En este ejemplo, vemos que $g(n) = f_{n-3} + 3$, por lo tanto:

$$f_n = f_{n-1} + f_{n-2} + 2f_{n-3} + 3, \underbrace{f_0 = 1,\ f_1 = 3,\ f_2 = 7}_{\text{Casos base}}, \underbrace{n > 2}_{\text{Dominio}}$$

In [6]:

```python
def f1(n):
    if n==0: return 1
    return 2*f1(n-1)+1

def f2(n):
    if n==0: return 1
    if n==1: return 3
    return 2*f2(n-2)+f2(n-1)+2

def f3(n):
    if n==0: return 1
    if n==1: return 3
    if n==2: return 7
    return f3(n-1)+f3(n-2)+2*f3(n-3)+3

for n in range(0, 15):
    print("%d \t %d \t %d." % (f1(n), f2(n), f3(n)))
```

```
1        1        1.
3        3        3.
7        7        7.
15       15       15.
31       31       31.
63       63       63.
127      127      127.
255      255      255.
511      511      511.
1023     1023     1023.
2047     2047     2047.
4095     4095     4095.
8191     8191     8191.
16383    16383    16383.
32767    32767    32767.
```

Las 3 fórmulas recurrentes nos producen la misma secuencia. Sin embargo, es preferible obtener una expresión cerrada para f_n.

Ejemplos de expresiones cerradas y no cerradas:

Expresión	Tipo
2^n	Cerrada
$n^2 + n + 1$	Cerrada
$f_n = f_{n-1} + f_{n-2}, \ldots$	No cerrada
$\sum_{i=1}^{n} i$	No cerrada
$\frac{1}{2} \cdot n \cdot (n-1)$	Cerrada

Las recurrentes son expresiones **no** cerradas.

Ahora procederemos a hallar una expresión **no** recurrente para f_n. Esto lo haremos usando la primera recurrente que hallamos.

$$f_n = 2f_{n-1} + 1,\ n > 0,\ f_0 = 1.$$

Para ejemplo usaremos el método de iteraciones.

Método de iteraciones

$$\begin{aligned}
k = 1.\ f_n &= 2 \cdot f_{n-1} + 1 \\
k = 2.\ f_n &= 2 \cdot \left(\underbrace{2 \cdot f_{n-2} + 1}_{f_{n-1}}\right) + 1 \\
&= 2^2 \cdot f_{n-2} + 2^1 + 2^0 \\
k = 3.\ f_n &= 2^2 \cdot \left(\underbrace{2 \cdot f_{n-3} + 1}_{f_{n-2}}\right) + 2^1 + 2^0 \\
&= 2^3 \cdot f_{n-3} + 2^2 + 2^1 + 2^0
\end{aligned}$$

Definimos entonces la *ecuación paramétrica*:

$$f(n, k) = 2^k \cdot f_{n-k} + 2^{k-1} + \cdots + 2^1 + 2^0$$

Recordemos la **serie geométrica:**

$$g(n,r,a) = \sum_{i=0}^{n-1} a \cdot r^i = a \cdot \frac{1-r^n}{1-r}$$

Por ejemplo

$$4 \cdot 3^0 + 4 \cdot 3^1 + 4 \cdot 3^2 + \cdots + 4 \cdot 3^9 = \sum_{i=0}^{n} a \cdot r^i =$$

Nota: El último expontente es $n-1$.
En este ejemplo $a = 4$, $r = 3$, $n = 10$.

$$4 \cdot 3^0 + 4 \cdot 3^1 + 4 \cdot 3^2 + \cdots + 4 \cdot 3^9 = \sum_{i=0}^{n} 4 \cdot 3^i = 4 \cdot \frac{1-3^n}{1-3} = 4 \cdot \frac{3^n - 1}{2} = 2 \cdot 3^n - 2.$$

In [15]:

```
def geom(n, a, r):
    # Expresión no cerrada (sumatori)
    sumatoria = 0
    i = 0
    while i < n:
        sumatoria += a*r**i
        i += 1

    # Expresión cerrada
    g = a*(1-r**n)//(1-r)

    return sumatoria, g

v1, v2 = geom(10, 4, 3)
print("Expresión no cerrada: %d, Expresión cerrada: %d." % (v1, v2))
```

```
Expresión no cerrada: 118096, Expresión cerrada: 118096.
```

Continuamos con nuestro ejemplo.

Recordemos que estábamos en la paramétrica de f_n, hallada luego de aplicar el método de iteraciones.

$$\begin{aligned} f(n,k) &= 2^k \cdot f_{n-k} + 2^{k-1} + \cdots + 2^1 + 2^0 \\ &= 2^k \cdot f_{n-k} + \sum_{i=0}^{k-1} 2^i \end{aligned}$$

Para *remover* el parámetro k usamos el (los) caso(s) base.

A partir del caso base: $n - k = 0 \implies n = k$ y por tanto: $f_{n-k} = f_{n-n} = f_0 = 1$.

Reemplazando esto en la paramétrica tenemos:

$$\begin{aligned} f_n &= 2^n \cdot f_0 + \sum_{i=0}^{n-1} 2^i \\ &= 2^n \cdot 1 + \sum_{i=0}^{n-1} 2^i \\ &= 2^n + \sum_{i=0}^{n-1} 2^i \end{aligned}$$

Ahora, evaluamos esta sumatoria utilizando la fórmula de la serie geométrica. En este caso, la utilizamos porque esta tiene la forma de ser una serie geométrica.

$$g(n, r, a) = \sum_{i=0}^{n-1} a \cdot r^i = a \cdot \frac{1 - r^n}{1 - r}$$

Para nuestro ejemplo, $r = 2$, $a = 1$. Entonces:

$$\begin{aligned} f_n &= 2^n + \underbrace{\sum_{i=0}^{n-1} 2^i}_{g(n,\, r=2,\, a=1)} \\ &= 2^n + \underbrace{\frac{1 - 2^n}{1 - 2}}_{g(n,\, r=2,\, a=1)} = 2^n + \frac{2^n - 1}{1} \\ &= 2^n + 2^n - 1 = 2^{n+1} - 1. \end{aligned}$$

Entonces, finalmente: $f_n = 2^{n+1} - 1$.

Nota: $2^n + 2^n = 2 \cdot 2^n = 2^{n+1}$.

In [17]:

```
def recurrente(n):
    if n==0: return 1 #f(0)=1
    return 2*recurrente(n-1)+1 #f(n)=2f(n-1)+1

def no_recurrente(n):
    return 2**(n+1)-1

for n in range(0, 10):
    print("Recurrente: %d,\t No recurrente: %d" % (recurrente(n), no_recurrente(n)))
```

```
Recurrente: 1,	 No recurrente: 1
Recurrente: 3,	 No recurrente: 3
Recurrente: 7,	 No recurrente: 7
Recurrente: 15,	 No recurrente: 15
Recurrente: 31,	 No recurrente: 31
Recurrente: 63,	 No recurrente: 63
Recurrente: 127,	 No recurrente: 127
Recurrente: 255,	 No recurrente: 255
Recurrente: 511,	 No recurrente: 511
Recurrente: 1023,	 No recurrente: 1023
```

Como se puede observar, ambas expresiones nos dan los mismos valores para un mismo n. Sin embargo, se requiere de una demostración formal de esto (Inducción matemática).

Ejercicios

- $f_n = 3 \cdot f_{n-1} + 2,\ n > 1,\ f_1 = 3$
- $f_n = 4 \cdot f_{n-1} + 5,\ n > 2,\ f_2 = 0$