

**Universidad de San Buenaventura**

**Facultad ingeniería de sistemas**



**UNIVERSIDAD DE  
SAN BUENAVENTURA  
CALI**

**Taller 3 Corte 2**

**Análisis de algoritmos**

**Presenta:**

Juan Felipe Hurtado Villani

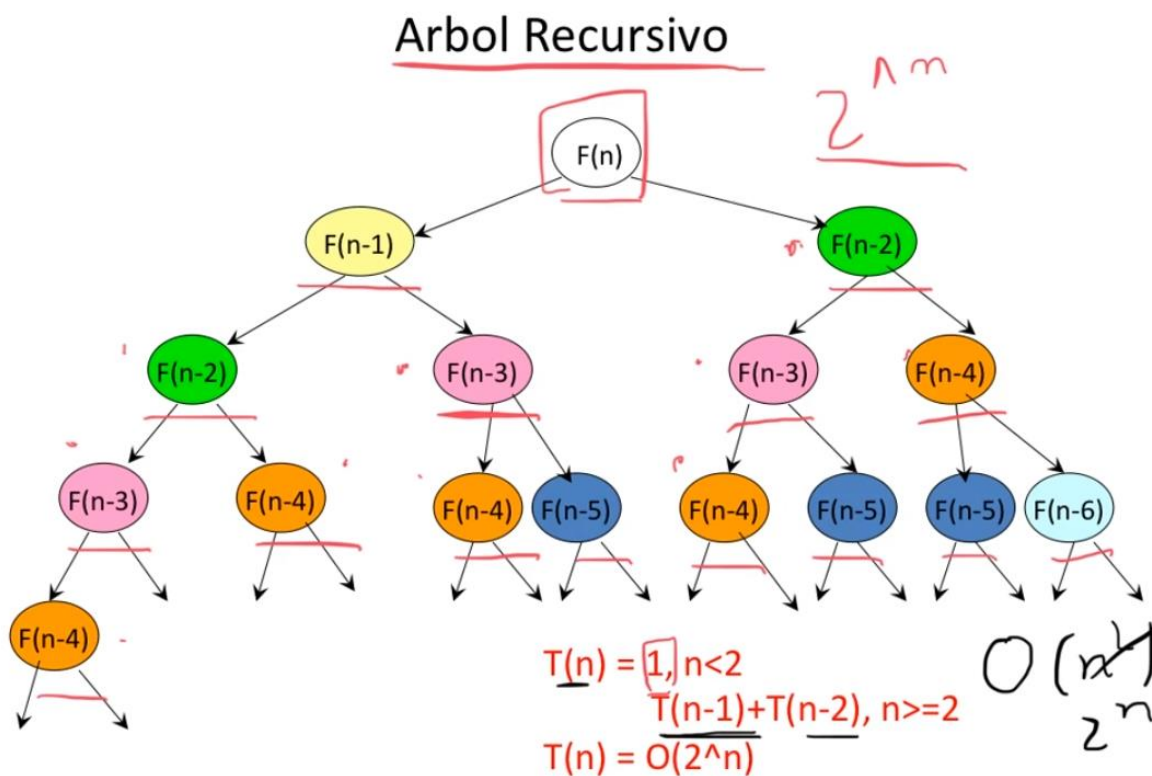
Cristian Apraez

Samuel Martínez

## PUNTO 1:

La forma recursiva de nuestro algoritmo, sería la siguiente:

```
1 def fib(n):  
2     if (n <=1):  
3         return 1  
4     else:  
5         return fib(n-1) + fib(n-2)
```

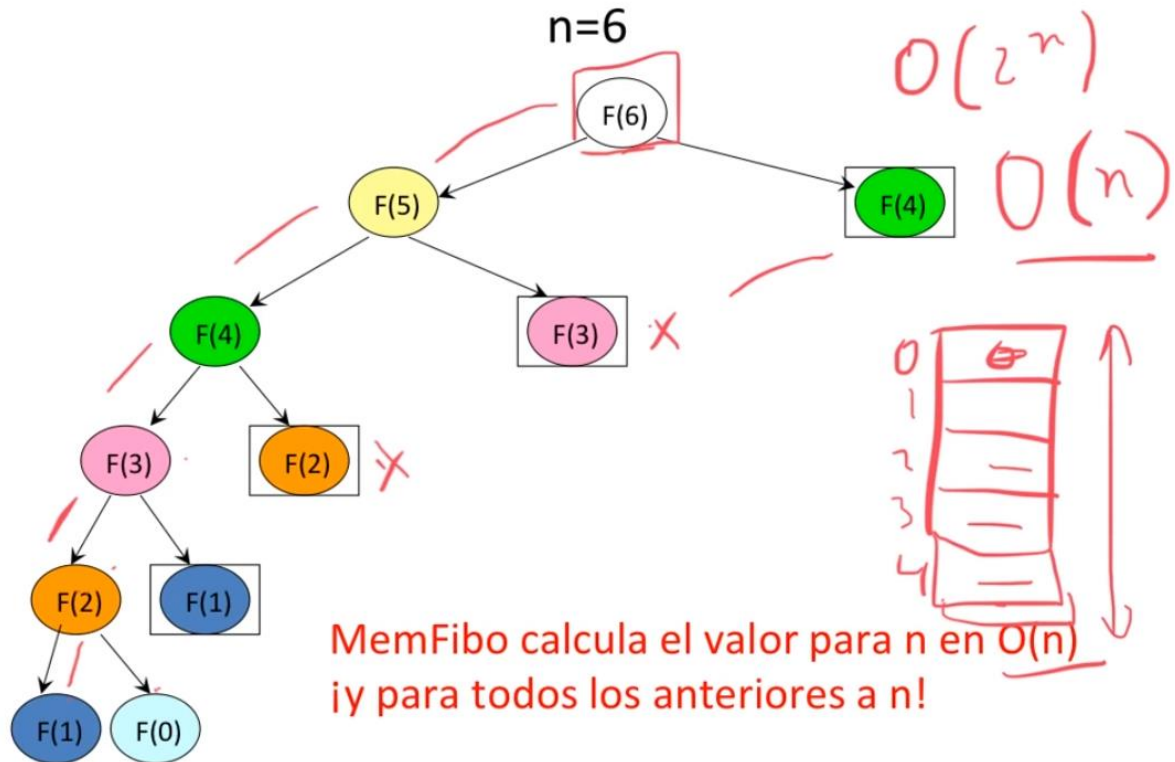


Esta sería la manera de hacerlo de manera recursiva.

Ahora, pasando a la manera dinámica, más específicamente al caso de Memorización.

Lo que nos dice es, que podemos mejorar la eficiencia del algoritmo recursivo si guardamos en memoria los resultados de las llamadas recursivas y los accedamos cuando los necesitamos luego:

## Calcular Recursivamente con Memoisación para



```
1 function MemFibo(n)
2   if (n < 2)
3     return n
4   else
5     if(not) F[n].done
6       F[n].value = MemFibo(n-1)+MemFibo(n-2)
7       F[n].done = true
8     return F[n].value
```

### PUNTO 2:

Para resolver este problema, lo que hicimos dividir el arreglo en 2 partes con unos bucles y luego hacer la sumatoria de ambas y dependiendo la que tuviera mayor pues se añadía al arreglo.

```

1  """El arreglo debe estar ordenado para el funcionamiento del algoritmo"""
2  def sumMaxima(arr,m):
3      n = 0
4      i = 0
5      a = 0
6      sumaSubArreglo=0
7      if len(arr)%2==0:
8          n = 0
9      else:
10         n = 1
11         lista = []
12         while n < len(arr):
13             lista.clear()
14             for x in range(len(arr)//2):
15                 sumaSubArreglo+=arr[n]
16                 lista.append(arr[n])
17                 n+=1
18             if sumaSubArreglo > a:
19                 a = sumaSubArreglo
20             sumaSubArreglo=0
21             if(m > len(arr)//2):
22                 if(m%2==0):
23                     if(len(arr)%2!=0):
24                         n=(len(arr)//2)
25                     else:
26                         n=(len(arr)//2)-1
27                     i = (len(arr))-m
28             else:
29                 if(len(arr)%2!=0):
30                     n=(len(arr)//2)
31                 else:
32                     n=(len(arr)//2)-1
33                     i = (len(arr))-m
34             while n>=i:
35                 a+=arr[n]
36                 n-=1
37             return a
38
39  print(sumMaxima([1,7,8,9,13,16,21],6))
40

```