

Tamara Nicole Rodríguez Luna

2021077818

Instituto Tecnológico de Costa Rica

IC-3002 Análisis de Algoritmos

Tarea 03 –  $T(n)$  y BigOh.

24 de agosto del 2023.

Profesor: José Helo.

## ————— Pregunta 1: Foo el Spaguetti.

```
foo(n):=block
( [t,x],           ---> 1
  t: 5 + (15*20),   ---> 1
  for x:1 thru n do ---> n
    print(x),         ---> 1
  );
  for x:1 thru n do ---> n
    for y:1 thru n do ---> n
      print(x*y)     ---> 1
);
Calcular T(n):
T(n) = 1 + 1+ n + n2
T(n) = 2 + n + n2
T(n) = n2 + n + 2
Calcular Big Oh:
O(n2)
```

## ————— Pregunta 2: Foo for for.

Observe que el segundo "for" depende del valor de "i".

Puede investigar cómo resolver este problema.

```
foo(n):=block
(
  for i:1 thru n do   ---> n
    for j:1 thru n-i do ---> (n-1)+(n-2)+...+2+1 =  $(\frac{1}{2}) * n * (n+1) - n = (\frac{1}{2}) * (n^2) - n$ 
      print(i,j)       ---> 1
);
```

Esto no es correcto

Calcular  $T(n)$ :  
 $T(n) = n$

$$T(n) = \frac{n \cdot (n+1)}{2} - n$$

$$T(n) = n \cdot \left( \frac{n \cdot (n+1)}{2} - n \right)$$

$$T(n) = n \cdot \left( \frac{n^2 + n}{2} - n \right)$$

$$T(n) = n \cdot \left( \frac{n^2}{2} + \frac{n}{2} - n \right)$$

$$T(n) = \frac{n^3}{2} + \frac{1n^2}{2} - n^2$$

$$T(n) = \frac{n^3}{2} - \frac{n^2}{2}$$

## Calcular Big Oh:

$$\text{Si } T(n) = \frac{n^3}{2} - \frac{n^2}{2}$$

$$O\left(\frac{n^3}{2}\right) = O(n^3 \cdot \frac{1}{2})$$

Entonces el cálculo final de Big Oh es  $O(n^3)$ .

El BigOh es incorrecto

— Pregunta 3: Foo llama a Foo.

foo(n):=block

( [r], ---> 1

6

```
print(n,float(n/2)),--> 1
```

r: foo( $n/2$ ),       $\longrightarrow T\left(\frac{n}{2}\right)$

$$1 + r \quad \longrightarrow 1$$

)

);

Calcular  $T(n)$ :

$$T(n) = T\left(\frac{n}{2}\right) + 1 + 1 + 1$$

$$T(n) = T\left(\frac{n}{2}\right) + 3$$

$$T(0) = 1$$

$$T(1) = 1$$

Con ayuda de Wolfram Alpha

$$T(0) = 1, T(1) = 1, T(n) = T\left(\frac{n}{2}\right) + 3$$

Resultado de Wolfram

$$T(n) = 3 \log n \text{ cuando } n \rightarrow \infty.$$

Calcular Big Oh:

$$O(\log n) \text{ cuando } n \rightarrow \infty.$$

— Pregunta 4: Foo Factorial.

```
foo(n):=block
```

```
(
```

```
if n=0 then ---> En este caso no tomaremos en cuenta el if para la asignación
```

```
1 asignación de costo, solamente el else ya que es la rama del  
if-then-else más pesada en el código presente.
```

```
else
```

```
n * foo(n-1) ---> 2T(n-1)
```

```
);
```

Calcular  $T(n)$ :

$$T(0) = 1$$

$$T(n) = n * T(n-1)$$

Con ayuda de Wolfram Alpha

$$T(0) = 1, T(n) = 2T(n-1)$$

Resultado de Wolfram

$$T(n) = 2^n$$

Calcular Big Oh:

$$O(2^n) \text{ cuando } n \rightarrow \infty$$

Lo que puso en las líneas del código no es cor

```
foo(5) = 5*foo(4)
        = 5*4*foo(3)
        = 5*4*3*foo(2)
        = 5*4*3*2*foo(1)
        = 5*4*3*2*1*foo(0)
        = 5*4*3*2*1*1
        = 5*4*3*2*1
        = 5*4*3*2
        = 5*4*6
        = 5*24
foo(5) = 120
```

---> Funcionamiento de la función factorial.

