Una visión intuitiva de la complejidad

Mauricio Toro

25 de mayo de 2016

Capítulo 1

Programa iterativo con una entrada

En este capítulo veremos los pasos para escribir un programa iterativo con una sola entrada, usualmente llamada "n"

1. *Escribir el programa o algoritmo*

```
for (int i = 1; i <= n; i++)
       for (int j = 1; j <= n; j++)
            print(i+"*"+j+"="+i*j);
```

2. *Etiquetar el programa*

```
for (int i = 1; i <= n; i++) // C1*n
       for (int j = 1; j <= n; j++) //C2*n2
```

```
print(i+"*"+j+"="+i*j); //C3*n2
```

3. Calcular el T(n) sumando todas las anotaciones

   $T(n) = (C2 + C3)n^2 + C1.n$

4. Aplicar la notación O

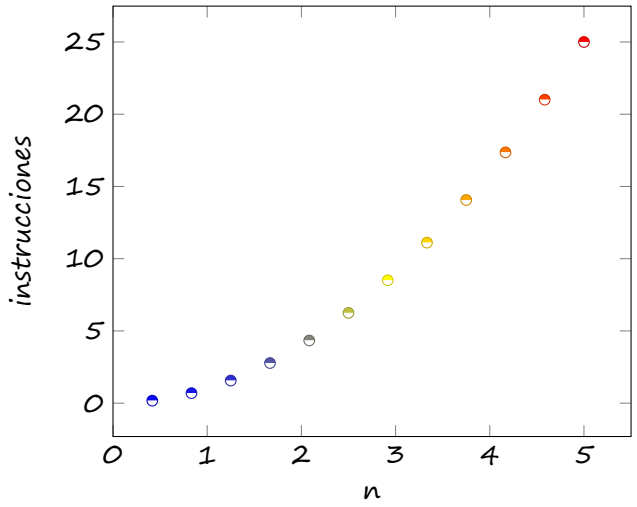
   $T(n)$ es $O((C2 + C3)n^2 + C1.n)$

5. Aplicar la regla de la suma

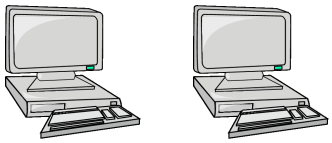
   $T(n)$ es $O((C2 + C3)n^2)$

6. Aplicar la regla del producto

   $T(n)$ es $O(n^2)$



Capítulo 2



Programa iterativo con varias entradas

En este capítulo veremos los pasos para escribir un programa iterativo con varias entradas, usualmente llamadas n, m o V, E.

1. *Escribir el programa o algoritmo*

```
for (int i = 1; i <= n; i++)
       for (int j = 1; j <= m; j++)
            print(i+"*"+j+"="+i*j);
```

2. *Etiquetar el programa*

```
for (int i = 1; i <= n; i++) // C1*n
       for (int j = 1; j <= m; j++) //C2*n.m
             print(i+"*"+j+"="+i*j); //C3*n.m
```

3. Calcular el T(n) sumando todas las anotaciones

   $T(n, m) = (C2 + C3)n.m. + C1.n$

4. Aplicar la notación O

   $T(n, m)$ es $O((C2 + C3)n.m + C1.n)$

5. Aplicar la regla de la suma

   $T(n, m)$ es $O((C2 + C3)n.m)$

6. Aplicar la regla del producto

   $T(n, m)$ es $O(n.m)$

Capítulo 3

Programa recursivo

En este capítulo veremos cómo calcular la complejidad de un programa recursivo.

1. *Escribir el programa o algoritmo*

```
SubProceso result <- Fibo( n )
   Definir result Como Entero;
   Si n <= 1 Entonces
        result <- n;
   Sino
        result<-Fibo(n-1)+Fibo(n-2);
```

2. *Etiquetar el programa*

```
SubProceso result <- Fibo( n )
   Definir result Como Entero; // C1
   Si n <= 1 Entonces // C2
        result <- n; //C3
   Sino
        result<-Fibo(n-1)+Fibo(n-2); //C4+T(n-1)+T(n-2)
```

3. Escribir el T(n) como una ecuación de recurencia

   $T(n) = C1 + C2 + C3$, si $n <= 1$

   $T(n) = C4 + T(n-1) + T(n-2)$, de lo contrario

4. Resolver la ecuación de recurrencia usando la teoría de polinomio característico, teorema maestro o https://www.wolframalpha.com/

   $T(n) = C * 2^n + C$

5. Aplicar la notación O

   $T(n)$ es $O(C * 2^n + C)$

6. Aplicar la regla de la suma

   $T(n)$ es $O(C * 2^n)$

7. Aplicar la regla del producto

   $T(n)$ es $O(2^n)$