

Práctico N°2

Ejercicio 13

Escribir en forma de algoritmo el método clásico para transformar un número natural de su representación binaria a su representación decimal. ¿Cuál es el costo, expresado en notación O de este algoritmo? Especifique las hipótesis que considere necesarias.

Ejemplo:

$$(1011)_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 8 + 0 + 2 + 1 = 11$$

Algoritmo el método clásico para transformar un número natural de su representación binaria a su representación decimal

```
Int binario_dec(in num_b)
```

```
    int resto, result, suma=0,i=0;
```

```
    if(num_b==1)
```

```
        return 1
```

```
    resto=num_b mod 10
```

```
    result= num_b div 10
```

```
    mientras (result>1)
```

```
        suma=suma+resto*2i
```

```
        resto=result mod 10
```

```
        result= result div 10
```

```
        i=i+1
```

```
    finmientras
```

```
    suma=suma+ resto*2i+ result *2(i+1);
```

```
    return suma;
```

0	1	2	3		
1	0	1	1		

¿Cuál es el costo, expresado en notación O de este algoritmo?

Int binario_dec(in num_b)

int resto, result, suma=0,i=0;

if(num_b==1) $\rightarrow O(1)$
return 1; $\rightarrow O(1)$

$O(O(1)+O(1)) =$
 $O(\max(O(1),O(1))) =$

resto=num_b mod 10 $\rightarrow O(1)$
result= num_b div 10 $\rightarrow O(1)$

$O(O(1)+O(1)) =$
 $O(\max(O(1),O(1))) =$

mientras (result>1) $\rightarrow O(n-1)$

suma=suma+resto*2^i; $\rightarrow O(1)$

resto=result mod 10 $\rightarrow O(1)$

result= result div 10 $\rightarrow O(1)$

i=i+1 $\rightarrow O(1)$

$O(O(1)+O(1)+O(1)+O(1)) =$
 $O(\max(O(1),O(1),$
 $O(1),O(1))) =$

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

finmientras

suma=suma+resto*2^i+ result
*2^(i+1); $\rightarrow O(1)$

return suma; $\rightarrow O(1)$

$O(O(1)+O(n-1)+O(1)+O(1)) =$
 $O(\max(O(1),O(n-1),O(1),O(1))) =$
 $O(n)$

Ejercicio 8

Decidir si son verdaderas o falsas las siguientes afirmaciones. Justificar su respuesta demostrando lo que afirma:

a) $3n^2 + 7n + 4$ es $O(n^2)$

Se puede resolver de dos maneras:

1- Por definición:

$O(f) = \{g/g \in F \wedge (\exists c \in \mathbb{R}^+) (\exists n_0 \in \mathbb{N}_0) (\forall n \geq n_0) (g(n) \leq c \cdot f(n))\}$ donde :

$$g(n) = 3n^2 + 7n + 4$$

$$f(n) = n^2$$

entonces

$$3n^2 + 7n + 4 \leq c \cdot n^2$$

$$(3n^2 + 7n + 4)/n^2 \leq c$$

$$3 + 7/n + 4/n^2 \leq c$$

En esta desigualdad cuando n crece, los últimos dos términos tienden a cero, entonces puedo encontrar una constante c donde a partir de un n_0 se cumpla siempre la desigualdad, por ejemplo:

$$N_0 = 1 \text{ y } c = 20$$

Encontrando un valor para n_0 y c queda demostrado que $g(n)$ pertenece al O de $f(n)$

2- Por límite

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} \begin{cases} \infty & \text{si } g \in O(f) \\ 0 & \text{si } f \in O(g) \\ \text{cte.} & \text{si } O(f) = O(g) \end{cases}$$

$$\lim_{n \rightarrow \infty} \left(\frac{n^2}{3n^2 + 7n + 4} \right) = 1/3 \text{ entonces } 3n^2 + 7n + 4 \in O(n^2)$$

Ejercicio 14

Expresar en notación O las siguientes recursiones:

a) $x(n) = x(n-1)+5$ para $n > 1$, 0 si $n = 1$

Desarrollo de la recursión:

$$1) x(n-1) + 5$$

$$2) (x(n-2)+5) + 5 = x(n-2)+10$$

$$3) (x(n-3)+10)+5 = x(n-3)+15$$

....

$$k) x(n-k)+5*k$$

La recursión termina cuando $n=1$, entonces

$$n-k=1 \Rightarrow k=n-1$$

Reemplazo k por $n-1$, dejando toda la ecuación en función de n

$$x(n-(n-1)+5*(n-1) = x(1)+5n-5$$

por regla de la suma $T1(n) + T2(n)$ es $O(\max(f,g))$.

Entonces $x(1)+5n-5$ es $O(\max(x(1),5n,5))$ es **$O(n)$**