Relación de ejercicios de Eficiencia - Tema 1

Alberto Llamas González

(1) Probar que las siguientes afirmaciones son verdad

1. 17 es $O(1)$

$$\lim_{x\to\infty} \frac{17}{1} = 17 < \infty \Rightarrow O(17) \subseteq O(1) \Rightarrow 17 \in O(1)$$

2. $\frac{n(n-1)}{2}$ es $O(n^2)$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{\frac{n(n-1)}{2}}{n^2} = \lim_{n\to\infty} \frac{\cancel{n}(n-1)}{2n^{\cancel{2}}} = \lim_{n\to\infty} \frac{n-1}{2n} =$$

$$= \frac{1}{2} \neq 0 \Rightarrow O\left(\frac{n(n-1)}{2}\right) = O(n^2) \text{ es decir}$$

$$\frac{n(n-1)}{2} \in O(n^2)$$

3. $\max(n^3, 10n^2)$ es $O(n^3)$

Por la regla de la suma. Si $T_1(n) = n^3 \in O(n^3)$ y $T_2(n) = 10n^2 \in O(n^2)$ entonces

$$T_1(n) + T_2(n) \text{ es } O(\max(n^3, 10n^2)) \Rightarrow$$

$$\Rightarrow O(n^3)$$

4. $\log_2(n)$ es $O(\log_3(n))$

Por las propiedades de los logaritmos sabemos que

$$\log_2(n) = \frac{\log(n)}{\log 2} \quad y \quad \log_3(n) = \frac{\log(n)}{\log 3}$$

Si aplicamos L'Hôpital

$$\lim_{n\to\infty} \frac{\log(n)/\log(2)}{\log(n)/\log(3)} = \lim_{n\to\infty} \frac{\log(3)}{\log(2)} \neq 0 \Rightarrow$$

$\Rightarrow$ Por tanto, como $O(\log_2(n)) \neq O(\log_3(n))$, no podemos afirmar que dicha sentencia sea cierta

(2) Encontrar el entero k más pequeño tal que $f(n)$ es $O(n^k)$ en los siguientes casos:

1. $f(n) = 13n^2 + 4n - 73$ Como $f(n) \in O(n^2) \Rightarrow k = 2$
2. $f(n) = 1/n+1$

$$\frac{1}{n+1} = \frac{1}{k} \Leftrightarrow n+1 = k \Leftrightarrow n = k-1$$ Como

sabemos que el orden de $f(n) \Rightarrow O(f(n)) = n$, $\underline{k = 1}$

3. $f(n) = \frac{1}{n-1}$

$$\frac{1}{n-1} = \frac{1}{k} \Leftrightarrow k = n-1 \Leftrightarrow n = k+1 \Rightarrow O(f(n)) = n,$$

es decir $f(n) \in O(n) \Rightarrow \boxed{k = 1}$

(2)

4. $f(n) = (n-1)^3 = n^3 - 3n^2 + 3n - 1 \in O(n^3)$

Sabemos que $f(n)$ es $O(n^3)$ luego $\boxed{k=3}$

5. $f(n) = \dfrac{n^3 + 2n - 1}{n+1}$

$$\lim_{n\to\infty} \frac{\left(\frac{n^3+2n-1}{n+1}\right)}{n^k} = \lim_{n\to\infty} \frac{\left(\frac{n^3+2n-1}{n+1}\right)}{n^k} \neq 0 \Rightarrow$$

$$\Rightarrow \frac{n^3+2n-1}{n+1} = n^k \Rightarrow O(f(n)) = O(n^k) \Rightarrow \boxed{k=3}$$

6. $f(n) = \sqrt{(n^2-1)}$

$$\lim_{n\to\infty} \frac{\sqrt{(n^2-1)}}{n^k} = \lim_{n\to\infty} \frac{(n^2-1)^{1/2}}{n^k} \neq 0 \Rightarrow k \geq 2 \Rightarrow \boxed{k=2}$$

Escaneado con CamScanner

③ Ordenar de menor a mayor las siguientes órdenes de eficiencia

$n, \sqrt{n}, n^3+1, n^2, n\log_2(n), 3^{\log_2(n)}, 3^n, 2^n+3^{n-1}, 20000, n+100, n2^n$

Cte Log Sublineal Lineales Superlineales

$20000 \subseteq \sqrt{n} \subseteq n \subseteq n+100 \subseteq n\log_2(n) \subseteq 3^{\log_2(n)} \subseteq n^2 \subseteq n^3+1 \subseteq$

Polinómicas

$\subseteq 3^n \subseteq 2^n+3^{n-1} \subseteq n2^n$

Exponenciales

④ Supongamos que $T_1(n) \in O(f(n))$ y $T_2(n) \in O(f(n))$.

Razonar la veracidad o falsedad.

1. $T_1(n) + T_2(n) \in O(f(n))$. Verdadero

Por la regla de la suma sabemos que si $T_1(n)$ es $O(f(n))$ y $T_2(n)$ es $O(g(n))$, entonces $T_1(n)+T_2(n)$ es $O(\max(f(n), g(n)))$. Como $g(n)=f(n) \Rightarrow O(\max(f(n), g(n)) = O(f(n))$. Por tanto

$$T_1(n) + T_2(n) \in O(f(n))$$

2. $T_1(n) \in O(f^2(n))$ Falso

Sea $T_1(n) = n^3$ y $f(n)=n$ donde $f^2(n) = n^2$

$T_1(n) = n^3 \notin O(n^2)$

3. $T_1(n)/T_2(n) \in O(1)$ Falso

Sea $T_1(n) = n^2$ y $T_2(n) = n$

$$\frac{n^2}{n} = n \in O(n) \text{ y } n \notin O(1)$$

Escaneado con CamScanner

(5) Considerar las siguientes funciones de n

1. $f_1(n) = n^2$

2. $f_2(n) = n^2 + 100n$

3. $f_3(n) = \begin{cases} n & \text{si } n \text{ es impar} \\ n^3 & \text{si } n \text{ es par} \end{cases}$

4. $f_4(n) = \begin{cases} n & \text{si } n \leq 100 \\ n^3 & \text{si } n > 100 \end{cases}$

Indicar para cada par distinto i, j si $f_i(n)$ es $O(f_j(n))$

Vemos que $f_1(n) \in O(n^2)$, $f_2(n) \in O(n^2)$, $f_3(n)$ no podemos compararla al variar en el ∞, $f_4(n) \in O(n^3)$ porque tiende a ella en el infinito.

Cogemos ahora los pares de funciones:

$f_1(n)$ es $O(f_2(n))$ y $f_1(n)$ es $O(f_4(n))$

$f_2(n)$ es $O(f_1(n))$ y $f_2(n)$ es $O(f_4(n))$

$f_4(n)$ NO es $O(f_1(n))$ y NO es $O(f_2(n))$

⑥ Obtener usando la notación O-mayúscula la eficiencia de ~~las~~ las siguientes funciones otrotos de código.

Código 1:

```
void ejemplo (n) {
    int i, j, k;
    for (int i=0 ; i<n; i++) {
        for (int j = i+1 ; j<n ; j++) {
            for (k = 0 ; k <= j ; k++)
                Globlal += k * i;
        }
    }
}
```

$\underbrace{\text{Globlal += k * i;}}_{O(1)}$] $O(n)$] $O(n^2)$] $O(n^3)$

La eficiencia del código 1 es $O(n^3)$

Código 2:

```
for (int i=0; i<n; i++)
    if (i % 2) {
        for (int j = i; j<n; j++)     ①
            x *= i;
        for (int j = 0 ; j<i; j++)    ②
            y *= j;
    }
```

③

En el if solo entra cuando i es par

① $\sum_{j=i}^{n-1} 1 = n-i$ ② $\sum_{j=0}^{i-1} 1 = i-1$ $\Rightarrow T_1(n) + T_2(n) = \underline{= n-1}$

③ $\sum_{i=0}^{n-1} n-1 = \sum_{i=0}^{n-1} n - \sum_{i=0}^{n-1} 1 = n^2 - n \in \boxed{O(n^2)}$

Código 3:

```
int funcion1 (int n){
    int s = 0;
    for(int i=0 ; i<n ; i++)
        s+=i;
    return s;
}
```

] $O(n)$ $\left[\sum_{i=0}^{n-1} 1 \ldots\right] \in O(n)$

```
int función2 (int n){
    int s = 0;
    for (int i=0; i<n; i++)
        s+= función1(i);
    return s;
}
```

] $O(n^2)$ porque tenemos que ~~múltiplicar~~ la eficiencia de la función 1 con la función 2, al llamarla $O(n) \cdot O(n) = \triangleright O(n^2)$

```
int main(){
    int k;
    cin>>k;
    cout << función2(k);
}
```

La eficiencia del código es $O(n^2)$