Algoritmos y Estructuras de Datos

Guía Práctica 5
Complejidad

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Ejercicio 1. Probar que las siguientes afirmaciones son verdaderas utilizando las definiciones.

a) $n^2 - 4n - 2 \in O(n^2)$.

b) Para todo $k \in \mathbb{N}$ y toda función $f : \mathbb{N} \to \mathbb{N}$, si $f \in O(n^k)$, entonces $f \in O(n^{k+1})$.

c) Si $f : \mathbb{N} \to \mathbb{N}$ es tal que $f \in O(\log n)$, entonces $f \in O(n)$.

Ejercicio 2. Determinar si cada una de las siguientes afirmaciones es verdadera o falsa. **Justificar** adecuadamente su respuesta.

a) $2^n \in O(1)$.

b) $n \in O(n!)$.

c) $n! \in O(n^n)$.

d) $2^n \in O(n!)$.

e) Para todo $i, j \in \mathbb{N}$, $i \cdot n \in O(j \cdot n)$.

f) Para todo $k \in \mathbb{N}$, $2^k \in O(1)$.

g) $\log n \in O(n)$.

h) $n! \in O(2^n)$.

i) $n^5 + bn^3 \in \Theta(n^5) \iff b = 0$.

j) Para todo $k \in \mathbb{R}$ $n^k log(n) \in O(n^{k+1})$.

k) Para toda función $f : \mathbb{N} \to \mathbb{N}$, $f \in O(f)$.

Ejercicio 3. ¿Qué significa, intuitivamente, $O(f) \subseteq O(g)$? ¿Qué se puede concluir acerca del crecimiento de f y g cuando, simultáneamente, tenemos $O(f) \subseteq O(g)$ y $O(g) \subseteq O(f)$?

Ejercicio 4. Responder, justificando adecuadamente su respuesta:

a) Sean f y g el mejor y el peor caso de un algoritmo. ¿Es cierto entonces que $g \notin O(f)$?

b) Sean $g(n)$ y $h(n)$ la cantidad de operaciones que realizan los algoritmos G y H en función del tamaño de la entrada. Si G ejecuta la mitad de operaciones que H, ¿vale que $g \in \Theta(h)$?

c) Un algoritmo que toma un arreglo como input realiza $\Theta(n^2)$ operaciones cuando el arreglo tiene más de 100 elementos y $\Theta(1)$ operaciones cuando tiene 100 o menos. ¿Cuál es el mejor caso del algortimo?

d) Si $f(n) < g(n), \forall n$, ¿es cierto que $f \notin \Omega(g)$?

e) Si la complejidad en el **peor caso** de un algoritmo es $\Omega(n)$, ¿es verdad que la complejidad de **mejor caso** no puede ser $O(1)$?

f) Si la complejidad temporal en el **peor caso** de un algoritmo pertenece a $O(n)$, entonces su complejidad temporal en el **mejor caso** también pertenece a $O(n^2)$.

Ejercicio 5. Indique si las siguientes afirmaciones son verdaderas o falsas. En caso de ser **verdadera**, justifique formalmente. En caso de ser **falsa**, dé un contraejemplo claro que respalde su análisis. Sean $f, g, h : \mathbb{N} \to \mathbb{N}$

1. Si $O(f(n)) \cap \Omega(g(n)) = \emptyset$, entonces $O(g(n)) \cap \Omega(f(n)) = \emptyset$.
2. Si $f \in O(g)$, entonces $f \in \Theta(g) \cup \Theta(h)$ para cualquier función h.
3. Si $f \in O(g)$ y $h \in O(g)$, entonces $(f + h) \in O(g)$
4. $O(n^2) \cap \Omega(n) = \Theta(n^2)$
5. $\Theta(n) \cup \Theta(n \log n) = \Omega(n \log n) \cap O(n)$
6. $f \in O(g) \iff O(f) \subseteq O(g)$
7. Si $f \in \Omega(g)$, entonces $O(f) \cap \Omega(g) = O(g) \cap \Omega(f)$

8. Si $f(n) < g(n)$ para todo n, entonces $\Theta(f)! = \Theta(g)$
9. Si $f \in O(g)$, entonces $f * g \in \Theta(g)$

Ejercicio 6. Determinar el orden de complejidad temporal de peor caso de los siguientes algoritmos, asumiendo que todas las operaciones sobre arreglos y matrices toman tiempo $O(1)$. La complejidad se debe calcular en función de una medida de los parámetros de entrada, por ejemplo, la cantidad de elementos en el caso de los arreglos y matrices y el valor en el caso de parámetros naturales.

a) SUMATORIA, que calcula la sumatoria de un arreglo de enteros:

```
function SUMATORIA(arreglo A)
    int i, total;
    total := 0;
    for i := 0 ... Long(A) - 1 do
        total := total + A[i];
    end for
end function
```

b) SUMATORIALENTA, que calcula la sumatoria de n, definida como la suma de todos los enteros entre 1 y n, de forma poco eficiente:

```
function SUMATORIALENTA(nat n)
    int i, total;
    total := 0;
    for i := 1 ... n do
        for j := 1 ... i do
            total := total + 1;
        end for
    end for
end function
```

c) PRODUCTOMAT, que dadas dos matrices A (de $p \times q$) y B (de $q \times r$) devuelve su producto AB (de $p \times r$):

```
function PRODUCTOMAT(matriz A, matriz B)
    int fil, col, val, colAFilB;
    matriz res(Filas(A), Columnas(B));
    for fil := 0 ... Filas(A) - 1 do
        for col := 0 ... Columnas(B) - 1 do
            val := 0;
            for colAFilB := 0 ... Columnas(A) - 1 do
                val := val + (A[fil][colAFilB] * B[colAFilB][col]);
            end for
            res[fil][col] := val;
        end for
    end for
    return res;
end function
```

d) INSERTIONSORT, que ordena un arreglo pasado como parámetro:

```
function INSERTIONSORT(arreglo A)
    int i, j, valor;
    for i := 0 ... Long(A) - 1 do
        valor := A[i];
        j := i - 1;
        while j ≥ 0 ∧ a[j] > valor do
            A[j+1] := A[j];
            j := j - 1;
        end while
        A[j+1] := valor;
    end for
end function
```

e) BúsquedaBinaria, que determina si un elemento se encuentra en un arreglo, que debe estar ordenado:

```
function BúsquedaBinaria(arreglo A, elem valor)
    int izq := 0, der := Long(A) - 1;
    while izq < der do
        int medio := (izq + der) / 2;
        if valor < A[medio] then
            der := medio;
        else
            izq := medio;
        end if
    end while
    return A[izq] = valor;
end function
```

Ejercicio 7. Para cada una de las siguientes afirmaciones, decida si son verdaderas o falsas y justifique su decisión.

a) $n + m \in O(nm)$.

b) $n + m^5 \in O(m^5)$.

c) $nm \in O(n + m)$.

d) $m^5 \in O(n + m^5)$.

Ejercicio 8. Sean $f, g : \mathbb{N} \to \mathbb{N}$, y supongamos que está definido el límite $\lim_{n\to+\infty} \frac{f(n)}{g(n)} = \ell \in \mathbb{R}_{\geq 0} \cup \{+\infty\}$. Probar que:

a) $0 < \ell < +\infty$ si y sólo si $f \in \Theta(g)$.

b) $\ell = +\infty$ si y sólo si $f \in \Omega(g)$ y $f \notin O(g)$.

c) $\ell = 0$ si y sólo si $f \in O(g)$ y $f \notin \Omega(g)$.

Recordar las definiciones de límite:

- $\lim_{n\to+\infty} a_n = \ell \in \mathbb{R}$ si $\forall\, \varepsilon > 0.\, \exists n_0 \in \mathbb{N}.\, \forall n > n_0.\, |a_n - \ell| < \varepsilon$.
- $\lim_{n\to+\infty} a_n = +\infty$ si $\forall\, M > 0.\, \exists n_0 \in \mathbb{N}.\, \forall n > n_0.\, a_n > M$.