Programación 3
Año 2020

Ejercicio resuelto en clase en semana 2

Ejercicio 1. [Kleinberg & Tardos, Ex. 2.6]

Tenemos un arreglo A de n números y queremos construir una matriz B, de dimensión $n \times n$, donde la entrada $B[i, j]$ tiene la suma de las entradas $A[i]$ hasta $A[j]$ para todo i, j, $1 \leq i < j \leq n$ (el contenido del resto de las entradas de B no está especificado). Para resolver el problema contamos con el algoritmo de la figura 1.1.

```
for i = 1 to n do
    for j = i + 1 to n do
        Sumar las entradas desde A[i] hasta A[j]
        Guardar el resultado en B[i, j]
    end
end
```

Figura 1.1: Algoritmo para el cálculo de la matriz B en el ejercicio 1.

(a) Dé explícitamente una función f tal que el tiempo de ejecución de este algoritmo es $\Theta(f(n))$; justifique su respuesta demostrando el resultado. **Sugerencia:** Observe que para cualquier constante positiva k y todo natural positivo n se cumple

$$\sum_{i=0}^{n-1} i^k \geq \sum_{i=\lceil n/2 \rceil}^{n-1} i^k \geq \left\lfloor \frac{n}{2} \right\rfloor \left(\frac{n}{2} \right)^k .$$

(b) Diseñe un algoritmo asintóticamente más eficiente. Muestre que el tiempo de ejecución de su algoritmo es $O(g(n))$, para cierta función g que debe especificar explícitamente, y demuestre que $g(n)$ es $o(f(n))$, es decir, $\lim_{n\to\infty} \frac{g(n)}{f(n)} = 0$.