Ejercicio 1 *(IzquierdaDominante)* ★

Escriba un algoritmo con dividir y conquistar que determine si un arreglo de tamaño potencia de 2 es *más a la izquierda*, donde "más a la izquierda" significa que:

- La suma de los elementos de la mitad izquierda superan los de la mitad derecha.
- Cada una de las mitades es a su vez "más a la izquierda".

Por ejemplo, el arreglo [8, 6, 7, 4, 5, 1, 3, 2] es "más a la izquierda", pero [8, 4, 7, 6, 5, 1, 3, 2] no lo es.

Intente que su solución aproveche la técnica de modo que complejidad del algoritmo sea estrictamente menor a $O(n^2)$.

Algoritmo

```python
def IzqDom(A):
    n = len(A)
    if n == 1: #O(1)
        return True

    mid = n//2 #O(1)
    izq = A[:mid] #O(n/2) ∈ O(n)
    der = A[mid:] #O(n/2) ∈ O(n)

    if sum(izq) > sum(der): #O(n/2 + n/2) = O(n)
        return IzqDom(izq) and IzqDom(der) #recursión
    else:
        return False
```

Complejidad

$$T(n) = 2T\left(\frac{n}{2}\right) + O(n)$$

Por teorema maestro:

$$\underbrace{\log_2 2}_{\log_b a} = \underbrace{1}_{c} \implies T(n) = \Theta\left(n^{\log_2 2} \log n\right) \in \Theta(n \log n)$$

Y alcanza con ver que $O(n \log n) \subset O(n^2)$

$$\lim_{n\to\infty} \frac{n \log n}{n^2} = \lim_{n\to\infty} \frac{\log n}{n} \underset{\text{LH}}{=} \lim_{n\to\infty} \frac{1}{n} = 0$$