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 $O(n^2)$.

2 4 6 8

8 6 | 7 4 | 5 1 | 3 2

Voy partiendo mi arreglo a la mitad hasta que el tamaño es $n=2$ → ahí me fijo si 1er elemento > 2do elemento → en ese caso además tengo que guardar la suma de los elementos de ese subarray para compararlo con el que corresponda

↪ necesito guardar un bool y una suma → tupla

↪ dividir y conquistar → si partimos usando índices → $\Theta(1)$

Falta combinar → pensemos en [8, 6, 7, 4]

caso base: [8,6] y [7,4] → tengo que fijarme que $8+6 > 7+4$
T T

Necesito que a partir de un bool devolver una suma → tupla <bool, nat>

↪ cuando comparo A[1] > A[2] pierdo la chance de sumar!

| I | D |

→ Para combinar me fijo si la tupla asociada al subarray I tiene true o false

→ si tiene true me fijo si D también. Si tiene me fijo que la suma de I > a la de D

→ si esto no sucede o D tiene false o I tiene false, no sumo nada y devuelvo false

```
<bool, nat> ID (arreglo A)
    if |A| = 2 then
        if A[0] > A[1] then                          Θ(1)
            return <true, A[0] + A[1]>               Θ(1)
        else
            return <false, 0>                        Θ(1)
    else
        A1 <- ID(A[0 : i/2])                         T(n/2)
        A2 <- ID(A[i/2 + 1, |A|])                    T(n/2)

        if A1.π1 ∧ A2.π1 then
            if A1.π2 > A2.π2 then
                return <true, A1.π2 + A2.π2>
            else
                return <false, -inf>
        else
            return <false, -inf>
```

Θ(1) son comparaciones de valores lógicos y sumas acotadas

→ Ec de recurrencia: $T(n) = 2T(n/2) + \underbrace{\Theta(1)}_{= d \;\; d \in \mathbb{R}_{>0}}$

c $d \in O(n^{\log_2 2 - \varepsilon})$ para algún $\varepsilon > 0$

$d \in O(n^{1-\varepsilon})$

→ Por caso 1 del TM → $T(n) = \Theta(n)$