

9. Calculá el orden del número de comparaciones del algoritmo del ejercicio 3.

Sea la función esta_ordenado definida en el ejercicio 3:

```
fun esta_ordenado(a: array[1..n] of nat) ret res: bool
  var i: nat

  i := 1
  res := true

  while i ≤ n-1 ∧ res = true do
    res := (a[i] ≤ a[i+1]) ∧ res
    i := i+1
  od
end fun
```

Calculo el número de comparaciones, es decir veo el ciclo de dicha función.

```
while i ≤ n-1 ∧ res = true do
  res := (a[i] ≤ a[i+1]) ∧ res
  i := i+1
od
```

Sea T el ciclo de dicha función, entonces:

$$\begin{aligned} \text{ops}(T) &= \text{ops}(\text{for } i := 1 \text{ to } n-1 \text{ do ... od}) \\ &= \sum_{i=1}^{n-1} \text{ops}(\text{res} := (a[i] \leq a[i+1]) \wedge \text{res}) \\ &= \sum_{i=1}^{n-1} 1 \\ \text{ops}(T) &= n - 1 \end{aligned}$$

∴ el orden de complejidad de la función esta_ordenado es **$O(n)$** o **lineal**.