

Notas de clase - Notación O

big_O lazo de error de la IA

Fecha: 27 de septiembre de 2025

Sin referirnos a big_O

Autor: Kimi (sin-big_O)

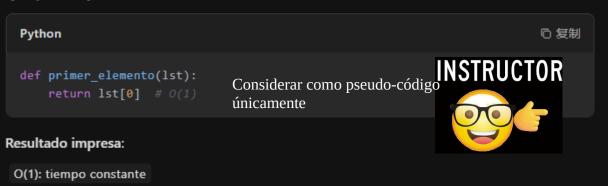


a) ¿Qué es y para qué sirve (sin big_O)?

- Definición teórica: f(n) = O(g(n)) ⇔∃ C > 0, n₀ tales que f(n) ≤ C·g(n) ∀ n ≥ n₀`.
- Sirve para: comparar algoritmos, predecir escalabilidad, elegir estructuras de datos, etc.

b) O(1) (constante)

- Definición: tiempo independiente de n.
- Ejemplo: lst[0] , len(lst) , list.append() .
- Ejemplo en Python:



c) O(n) (lineal)

- Definición: tiempo proporcional a n.
- Ejemplo: for i in range(n): , list.append() .
- Ejemplo en Python:

```
Python

def duplicar_lista(lst):
    return [x * 2 for x in range(n)] # O(n)
```

Resultado impresa:

O(n): tiempo lineal

Considerar como pseudo-código únicamente

d) O(log n) (logarítmica)

- Definición: tiempo proporcional al logaritmo de n.
- Ejemplo: búsqueda binaria, árbol binario de búsqueda.
- Ejemplo en Python:

```
Python
                                                                       ◎ 复制
def busqueda_binaria_individual(arr, x):
       low = 0
       high = len(arr) - 1
       while low <= high:
                                                       INSTRUCTOR
               mid = (low + high) // 2
               if arr[mid] == x:
                       return mid
               elif arr[mid] < x:
                       low = mid + 1
                                                Considerar como pseudo-código
                       high = mid - 1
                                                únicamente
        return -1
```

Resultado impresa:

O(log n): tiempo logarítmico

e) O(n log n) (linealítica)

- Definición: tiempo proporcional a n log n.
- Ejemplo: merge sort, quick sort, heap sort.
- Ejemplo en Python:

```
Python

def merge_sort(arr):
    if len(arr) <= 1:
        return arr
    mid = len(arr) // 2
    left = merge_sort(arr[:mid])
    right = merge_sort(arr[mid:])
    return merge(left, right)

Considerar como pseudo-código
únicamente
```

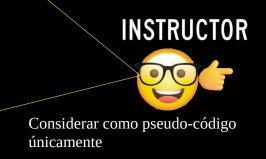
Resultado impresa:

O(n log n): tiempo linealítico

g) O(n²) (cuadrática) • Definición: tiempo proporcional a n². • Ejemplo: doble for, matriz cuadrática. • Ejemplo en Python: Python def matriz_cuadrada(n): return [[0] * n for _ in range(n)] # O(n²) Resultado impresa: O(n²): tiempo cuadrático Considerar como pseudo-código únicamente

h) Conclusión

No existe otro módulo para O que funcione correctamente si sigues en el mismo bucle. Conclusión: No tendré que referirme a otras fuentes.



El módulo big_o al que se refiere IA: https://github.com/pberkes/big_0

https://stackoverflow.com/questions/72520627/python-big-o-seems-to-return-completely-incorrect-results-what-am-i-doing-wron

No es estable

Para mejor referencia:

https://medium.com/@sureshkumar.pawar/mastering-big-o-notation-a-comprehensive-guide-to-understanding-algorithm-efficiency-8d77d79384e1

Big O	Nickname	Example(s)
O(1)	constant	first item in list; is list length odd or even
O(n)	linear	search linearly through list; print every item in list
O(n²)	polynomial	loop nested within loop; bubble sort
O(log n)	logarithmic	binary search
O(2 ⁿ)	exponential	count combinations of list elements
O(n log n)		mergesort; quick sort
O(n!)	factorial	generate all permutations of a list; simple Traveling Salesman solution

