



Notas de clase – Notación O

Autor: Kimi (sin-big_O)

Fecha: 27 de septiembre de 2025

Sin referirnos a big_O

big_O lazo de error de la IA

INSTRUCTOR



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

- Definición teórica: $f(n) = O(g(n)) \Leftrightarrow \exists C > 0, n_0$ tales que $f(n) \leq C \cdot g(n) \forall n \geq n_0$.
- 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:

Python

复制

```
def primer_elemento(lst):  
    return lst[0] # O(1)
```

Considerar como pseudo-código únicamente

INSTRUCTOR



Resultado impresa:

`O(1): tiempo constante`

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)] #  $O(n)$ 
```

INSTRUCTOR



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:  
        mid = (low + high) // 2  
        if arr[mid] == x:  
            return mid  
        elif arr[mid] < x:  
            low = mid + 1  
        else:  
            high = mid - 1  
    return -1
```

INSTRUCTOR



Considerar como pseudo-código únicamente

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) $O(n^2)$ (cuadrática)

- **Definición:** tiempo proporcional a n^2 .
- **Ejemplo:** doble for, matriz cuadrática.
- **Ejemplo en Python:**

Python

复制

```
def matriz_cuadrada(n):  
    return [[0] * n for _ in range(n)] #  $O(n^2)$ 
```

Resultado impresa:

$O(n^2)$: tiempo cuadrático

Considerar como pseudo-código únicamente

INSTRUCTOR



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.

INSTRUCTOR



Considerar como pseudo-código únicamente

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

<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
$O(n^2)$	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

Horrible Bad Fair Good Excellent

