

```
ExamenEjemplo3y4_LaraMartinezChristianGael.py X ExamenEjemplo1y2_LaraMartinezChristianGael.py
C: > Users > chris > OneDrive > Escritorio > Examen_LaraMartinezChristianGael > ExamenEjemplo3y4_LaraMartinezChristianGael.py

1  def max_value_after_operations(n, queries): →  $O(n)$ 
2      arr = [0] * (n + 1) →  $n+1$ 
3
4      for query in queries:
5          a, b, k = query
6          arr[a] += k →  $O(n)$ 
7          if b + 1 <= n:
8              arr[b + 1] -= k
9
10     max_value = 0
11     current_value = 0
12
13     for value in arr:
14         current_value += value
15         max_value = max(max_value, current_value)
16
17     return max_value
18
19 # Ejemplo
20 n = 10
21 queries = [[1, 5, 3], [4, 8, 7], [6, 9, 1]]
22 result = max_value_after_operations(n, queries)
23 print(result) # Salida esperada: 10
```

$O(n^2)$

La notación Big O del algoritmo es  $O(n^2)$ .

Donde “n” es el tamaño del arreglo. Esto se debe a que el algoritmo itera sobre el arreglo dos veces

- una vez para agregar los valores k a los elementos del arreglo.
- otra vez para encontrar el valor máximo.

La notación Big O del algoritmo de segmentación es  $O(n)$ , donde **n es el tamaño del arreglo.**

El algoritmo solo itera sobre el arreglo una vez, para actualizar el valor máximo en cada segmento.

Por lo tanto, el algoritmo de segmentación es más eficiente que el algoritmo original en un factor de n.