

UNIVERSIDAD DEL VALLE DE GUATEMALA

Teoría de la computación

Sección 20

Catedrático: Gabriel Brolo



Laboratorio 8

Teoría de la computación

Gonzalo Enrique Santizo Vega - 21504

GUATEMALA, 4 de octubre de 2023

Ejercicio 1:

El bucle exterior se ejecutará desde $n // 2$ hasta n . Esto implica aproximadamente $(n - n // 2) + 1$ iteraciones, lo cual es del orden de n .

El bucle interno se ejecutará desde 1 hasta $n - (n // 2) + 1$, lo cual es del orden de n .

Dentro del bucle interno, tienes un bucle while que se ejecuta mientras k sea menor o igual a n , donde k se duplica en cada iteración. Esto significa que el bucle while se ejecutará hasta que k sea igual o mayor que n , lo cual ocurre cuando k alcanza o supera n en la forma de una potencia de 2. En otras palabras, se ejecutará hasta que k llegue a ser mayor o igual a n , lo que sucede en aproximadamente $\log_2(n)$ iteraciones.

En resumen, el número total de operaciones fundamentales se puede aproximar como el producto de las iteraciones de los tres bucles:

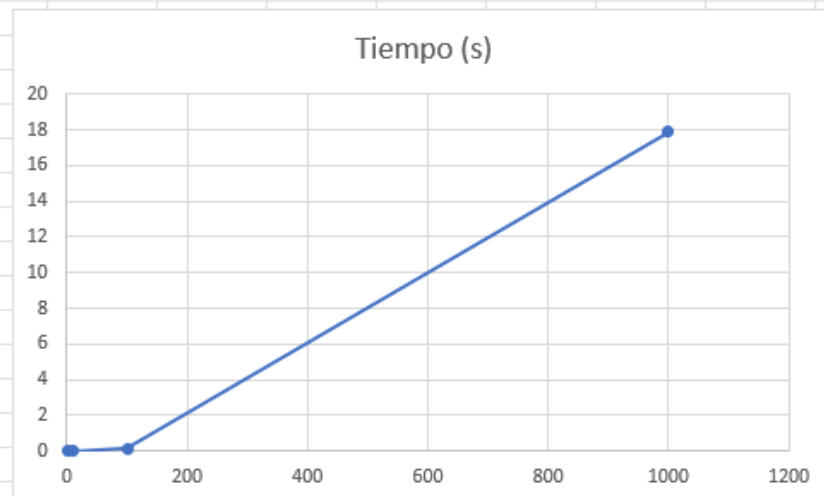
Bucle exterior: $O(n)$

Bucle interno: $O(n)$

Bucle while: $O(\log n)$

Entonces, el tiempo de ejecución total es $O(n * n * \log n)$, que se puede simplificar a $O(n^2 * \log n)$ en notación Big O. Por lo tanto, la complejidad es $O(n^2 * \log n)$ en el peor caso

Input	Tiempo (s)
1	0
10	0
100	0.13
1000	17.89



Ejercicio 2:

$O(1)$ o tiempo constante, ya que el número de operaciones fundamentales es fijo y no depende del tamaño de entrada n . El tiempo de ejecución será aproximadamente el mismo sin importar el valor de n

Alg 2	
Input	Tiempo (s)
1	0
10	0
100	0.26
1000	3.87
10000	3.88
100000	35.08
1000000	392.02



Ejercicio 3:

El bucle exterior se ejecutará desde 1 hasta $(n // 3)$, donde n es el tamaño de entrada. Esto implica aproximadamente $n / 3$ iteraciones, lo cual es del orden de n .

El bucle interno se ejecutará desde 1 hasta n , con un paso de 4 en cada iteración. Esto significa que el bucle interno se ejecutará en $(n / 4)$ iteraciones.

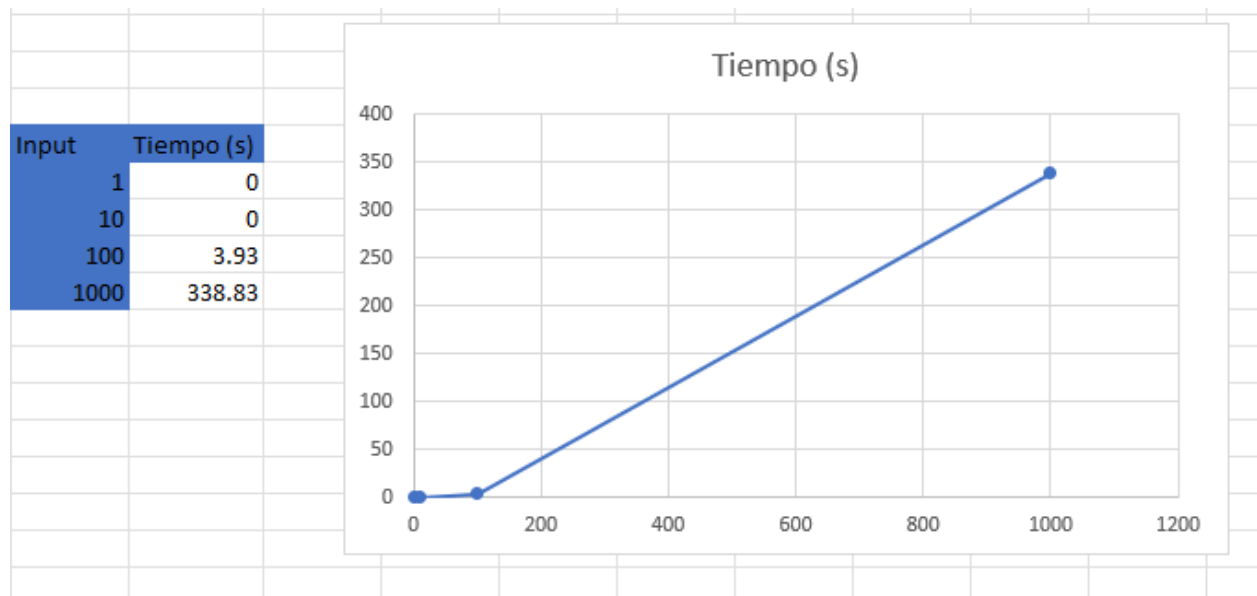
Dentro de estos bucles, tienes una impresión de "Sequence". Sin embargo, la impresión es una operación constante, es decir, no depende de n .

Entonces, el número total de operaciones fundamentales en este algoritmo se puede aproximar como el producto de las iteraciones de los dos bucles:

Bucle exterior: $O(n)$

Bucle interno: $O(n)$

Como ambos bucles son lineales en relación con n , la complejidad del algoritmo es $O(n * n)$, que se simplifica a $O(n^2)$



Ejercicio 4:

Mejor Caso:

El mejor caso ocurre cuando el elemento que estamos buscando se encuentra en la primera posición del arreglo. En este caso, solo se necesita una comparación para encontrar el elemento. El tiempo de ejecución en el mejor caso es constante y se puede representar como $O(1)$.

Caso Promedio:

El caso promedio se refiere a una situación en la que el elemento que estamos buscando tiene una probabilidad igual de encontrarse en cualquier posición del arreglo. Para analizar el caso promedio, consideramos que cada posición tiene una probabilidad de $1/n$ de contener el elemento que buscamos (donde n es el tamaño del arreglo). Dado que en promedio recorreremos la mitad de la lista antes de encontrar el elemento (ya que la posición deseada es igualmente probable en cualquier lugar), el tiempo de ejecución en el caso promedio es $O(n/2)$, lo que se reduce a $O(n)$.

Peor Caso:

El peor caso ocurre cuando el elemento que estamos buscando está en la última posición del arreglo o no se encuentra en el arreglo en absoluto. En este caso, debemos recorrer todo el arreglo hasta el final para determinar que el elemento no está presente. El tiempo de ejecución en el peor caso es $O(n)$, donde n es el tamaño del arreglo.

Ejercicio 5:

a