Ejercicios de Análisis de Algoritmos

# Ejercicios

## Análisis de Algoritmos

Resuelva los siguientes ejercicios:

1. El algoritmo A tarda 5*n*2 en resolver un determinado problema con n datos de entrada, mientras que B tarda *n*2 + 400*n*. ¿Cuál de los es más eficiente en relación al tamaño de la entrada?

En A y B, el término dominante es \(n^2\). Igualando las ecuaciones, encontramos que:

- Para n < 100 T\_B(n) podría ser mayor debido al término lineal significativo.

- Para n > 100, T\_A(n) será mayor debido al coeficiente más grande en el término cuadrático.

1. Un algoritmo toma 10 segundos en resolver un problema de tamaño N = 50. Si el algoritmo es cúbico ¿cuánto tardará para resolver un problema de N = 100?

Un algoritmo cúbico toma 10 segundos para N = 50. Dado que T(50) = k \* 50^3 = 10 segundos, encontramos k = 8 \* 10^-5.

Para N = 100

T(100) = k \* 100^3 = 80 segundos

1. Los algoritmos pueden ser eficientes para determinados tamaños de N. Existe el algoritmo A que tarda *n*2 + 200*n* y el algoritmo B que tarda *n*3 + *n*. Responda las siguientes preguntas:
   1. Determine el valor de N donde ambos algoritmos son igualmente eficientes.
   2. Si N es menor que el valor encontrado, ¿cuál algoritmo debería de utilizar? ¿Por qué?

3.a. Igualamos n2+200n=n3+nn^2 + 200n = n^3 + nn2+200n=n3+n para encontrar el punto donde ambos algoritmos son igualmente eficientes. Al resolver la ecuación, obtenemos que el punto de equilibrio es aproximadamente N=14.615N = 14.615N=14.615.

3.b. Si NNN es menor que 14.615, se debe utilizar el algoritmo A, ya que el término n3n^3n3 en el algoritmo B dominará rápidamente, haciéndolo menos eficiente.

4.

El bucle exterior se ejecuta fila veces, cada iteración del bucle exterior hace que el bucle interior se ejecute columna veces, la asignación dentro del bucle interior toma un tiempo constante O(1).

Por lo tanto, el tiempo total T(N) es el producto de las iteraciones de los dos bucles:

T(N) = filas \* columnas

El tiempo de ejecución asintótico es O(filas \* columnas).

5.

El primer bucle se ejecuta n veces, con cada iteración tomando un tiempo constante de 1.

El segundo bucle también se ejecuta n veces, y cada iteración toma un tiempo constante de 1.

Sumando los tiempos de ambos bucles, el tiempo total de ejecución es O(2n), que simplificamos a O(n)

Por lo tanto, el algoritmo tiene una complejidad de O(n).

T(n) = O(n).

**Búsquedas**

Preguntas a responder al terminar la actividad:

¿Puede estimar el costo O de la parte del algoritmo que hace las búsquedas?

La complejidad temporal del código es:

* Generación del Vector:O(n)
* Búsqueda Secuencial: 0(N^2)
* Búsqueda Binaria: O(NlogN)

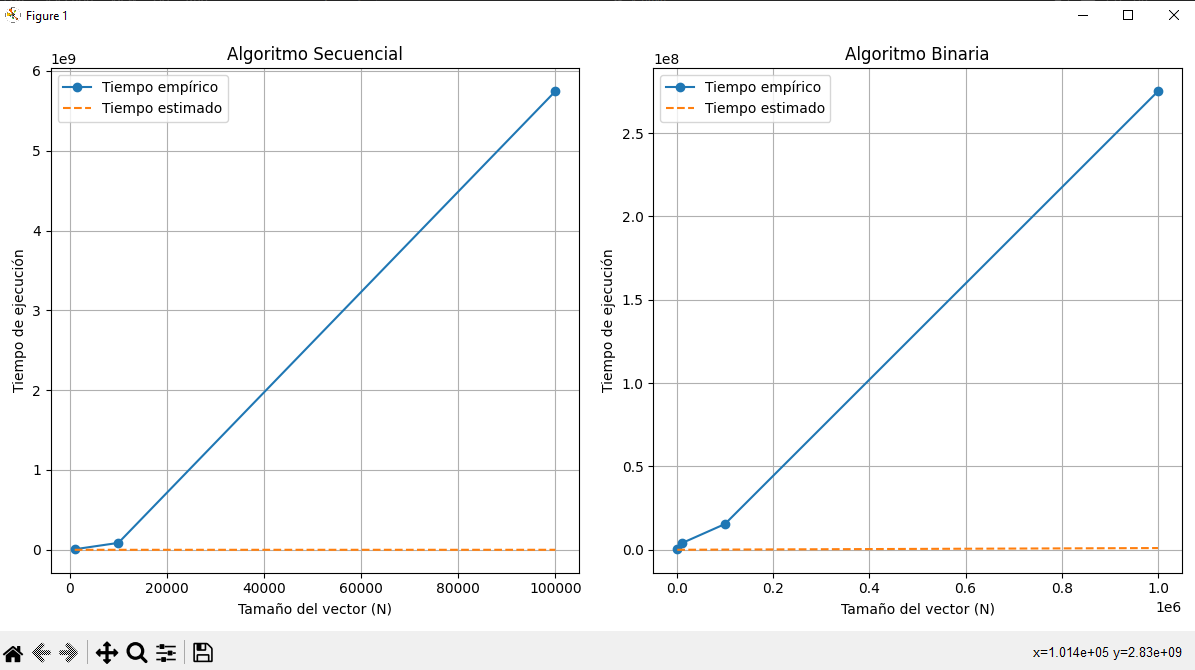
La complejidad de la búsqueda secuencial es cuadrática O(N^2) porque se realizan N búsquedas lineales en un vector de tamaño N

La complejidad de la búsqueda binaria es O(N log N) porque, aunque se realizan N búsquedas, cada una tiene una complejidad logarítmica gracias a la búsqueda binaria en un vector ordenado.

1. ¿Coinciden los tiempos observados con la estimación realizada?

Los tiempos observados no coinciden exactamente con la estimación realizada, pero siguen una tendencia similar de crecimiento.

3. ¿Puede generar un gráfico que muestre la evolución del tiempo de ejecución?



Secuencial

|  |  |  |  |
| --- | --- | --- | --- |
| N | Tiempo empírico | Tiempo estimado | Relación empírico/estimado |
| 1000 | 6701000 | 1,024 | 670,100 |
| 10000 | 86711900 | 16,384 | 6,193,707 |
| 100000 | 5744767300 | 131,072 | 337,929,252 |
| 1000000 | No finaliza | // | // |

Binaria

|  |  |  |  |
| --- | --- | --- | --- |
| N | Tiempo empírico | Tiempo estimado | Relación empírico/estimado |
| 1000 | 652400 | 1,024 | 65,240 |
| 10000 | 3932100 | 16,384 | 280,864 |
| 100000 | 15462700 | 131,072 | 909,567 |
| 1000000 | 275 134 300 | 1,048,576 | 13,756,715 |

Ordenacion

|  |  |  |  |
| --- | --- | --- | --- |
| N | Inserción | Merge Sort | Quick Sort |
| 10 000 | 60400 | 34000 | 27000 |
| 100 000 | 295200 | 271500 | 273900 |
| 1 000 000 | 2968200 | 3379600 | 2793500 |
| 10 000 000 | 25997500 | 30514800 | 25092300 |