



Introducción al análisis de algoritmos

Contenido

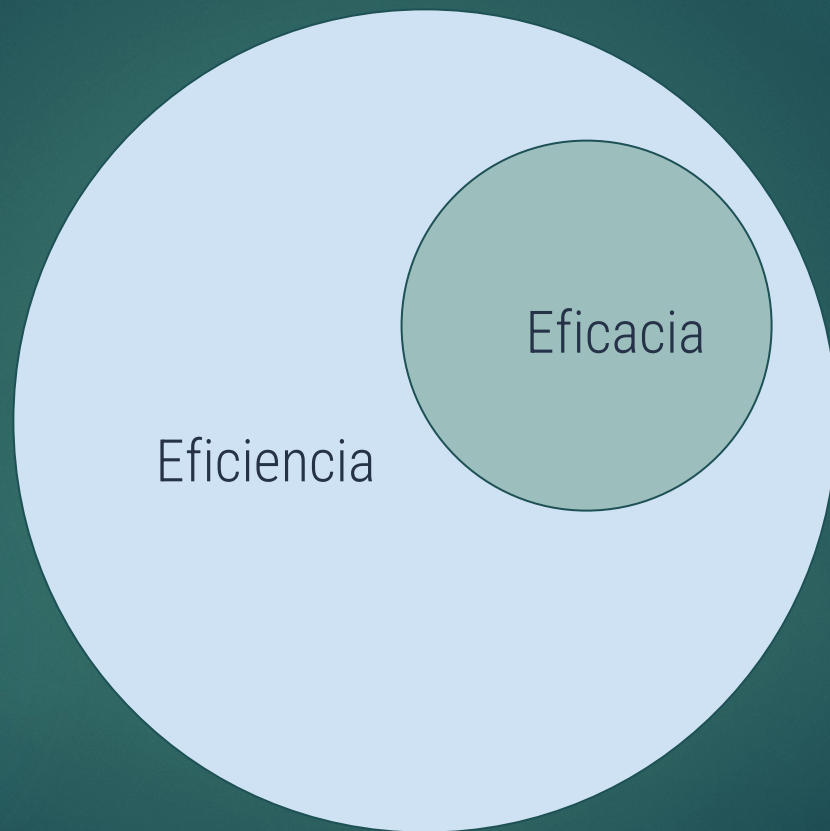
1. Eficiencia de un algoritmo
 - a. Eficiencia vs Eficacia
2. Complejidad temporal
3. Complejidad espacial
4. Complejidad de un algoritmo
 - a. Función de complejidad
5. Análisis asintótico
 - a. Big O (O)
 - b. Big Omega (Ω)
 - c. Big Theta (Θ)
6. Simplificación de función de complejidad
7. Órdenes más comunes



1

Eficiencia

Eficiencia vs Eficacia




Eficacia

- Capacidad para producir el efecto deseado o esperado.
- Se aplica a las cosas o personas que pueden producir el efecto o prestar el servicio a que están destinadas.

Eficiencia

- Capacidad de alcanzar los objetivos y metas programadas con el mínimo de recursos y tiempo disponibles, logrando su optimización.
- Capacidad para realizar o cumplir adecuadamente una función.



*¿Cuál es la diferencia
entre eficiencia y
eficacia?*

Eficiencia de un algoritmo

- Un problema puede ser resuelto por métodos distintos, cada uno con su grado de eficiencia.

Eficiencia de un algoritmo

- Es la cantidad de recursos utilizados por un algoritmo
- Los recursos a medir son:
 - Tiempo
 - Memoria

Eficiencia de un algoritmo

- ¿Deberíamos evaluar ambos simultáneamente?

Eficiencia de un algoritmo

- ¿Deberíamos evaluar ambos simultáneamente?
- Es necesario conocer cuál es el recurso más importante,

Eficiencia de un algoritmo

- ¿Deberíamos evaluar ambos simultáneamente?
- Es necesario conocer cuál es el recurso más importante,
- Un algoritmo es eficiente si puede ser ejecutado con un nivel aceptable de recursos



2

Complejidad Temporal

Conceptos

Complejidad Temporal



Complejidad temporal (T)

- Cuantifica la cantidad de tiempo tomado por un algoritmo para ejecutarse como función de una entrada de tamaño n .
- Mide el número de instrucciones elementales ejecutadas, el tiempo real depende de la rapidez en que es ejecutada una instrucción elemental.



3

Complejidad Espacial


Conceptos

Complejidad Espacial



Complejidad Espacial (E)

- Cuantifica la memoria necesaria para ejecutar un algoritmo con una entrada de tamaño n .
- Cantidad de variables y su respectivo espacio



*La complejidad espacial,
en general, tiene mucho
menos interés. El tiempo
es un recurso mucho más
valioso que el espacio.*



4

Complejidad de un Algoritmo

Función de complejidad



*La complejidad no es un
número: es una función.*

Función de complejidad

- En lugar de medir tiempos, se cuentan las instrucciones del algoritmo. Supondremos que cada instrucción se ejecuta en un tiempo constante.
- Esto es posible porque queremos saber el crecimiento en instrucciones necesarias para resolver el problema con respecto al tamaño de la entrada.

Función de complejidad

- Ambos tipos de complejidad pueden ser representados por funciones matemáticas:
 - n es la variable independiente
 - T/E son las variables dependientes
- Existen tres funciones por algoritmo para los casos mejor, peor y promedio. Éstas se analizan evaluando sus valores de salida cuando n tiende a infinito.

Ejemplo

$$T(n) = 4n + 4$$

```
1 var total = 0; //1
2 for (var i = 0; i < n; i++) { //1 + (1 + 1) * n = 2n + 1
3     total = total + i; // (1 + 1) * n = 2n
4 }
5
6 Console.WriteLine(total.ToString()); //2
7 //4n + 4
```

Ejemplo

$$T(n) = 5n^2 + 3n + 2$$

```
1 var x = 0; //1
2 for (var i = 0; i < n; i++) { //1+(1+1) * n = 2n + 1
3     for (var j = 0; j < n; j++) { //1+(1+1) * n * n = 2n^2 + n
4         x = x + 1; // (1+1) * n * n = 2n^2
5         Console.WriteLine(x); //n*n = n^2
6     }
7 }
8 //5n^2 + 3n + 2
```

¡Gracias por su
atención!