

Sesión 18 (S)

Complejidad de los algoritmos y notación asintótica.

When your interviewer asks for the time complexity of your algorithm but you have no idea what that means

DaCobalt • 1d
Big Oof notation

...

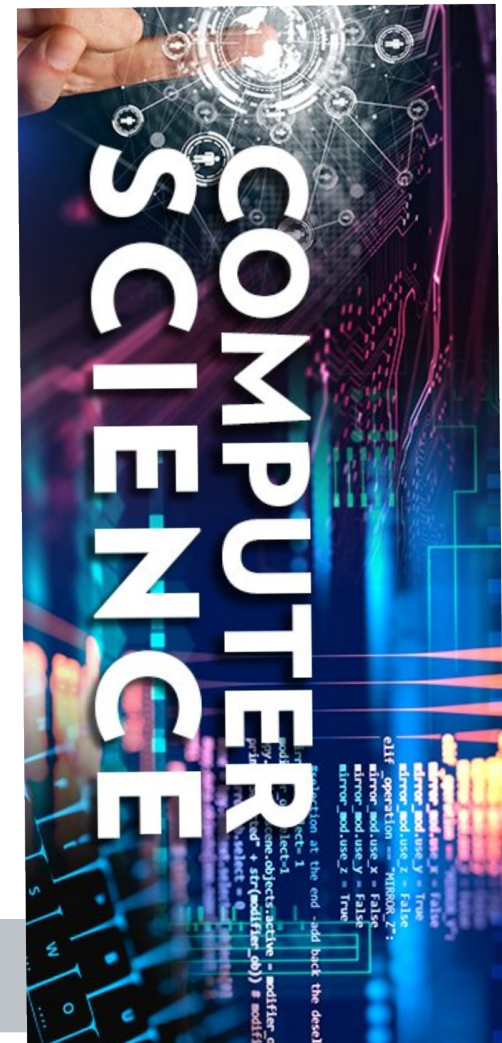
Reply



12



True story



Algoritmo

“Un algoritmo es un procedimiento para resolver un problema cuyos pasos son concretos y no ambiguos. El algoritmo debe ser correcto, de longitud finita y debe terminar para todas las entradas”.

$$y = f(x)$$

donde:

“y” es la salida

“x” los datos de entrada

“f(x)” el algoritmo que depende de la entrada

Eficacia y eficiencia

Eficacia => Propiedad inherente al algoritmo
(dar con la solución correcta).

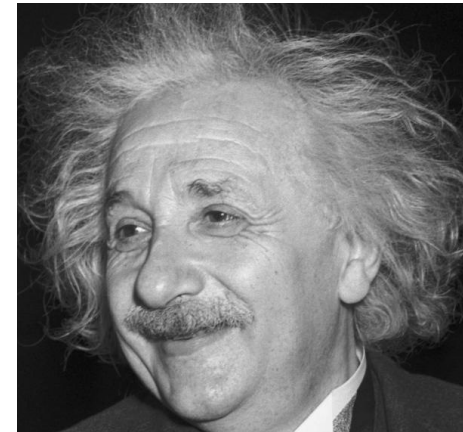


Eficiencia => Uso óptimo
de recursos.

Recursos

+ Tiempo de procesamiento

+ Uso de memoria



Algunos algoritmos son eficientes y otros no.
¿Cómo medir la eficiencia de un algoritmo?



**“MATEMÁTICAS
HIJO,
LOS NÚMEROS
NO MIENTEN”**

Pensamos acerca del tiempo de ejecución del algoritmo como una función del tamaño de su entrada.



```
int arr1[] = {1,3,4,4};  
int arr2[] = {1,2,4,1,2,5,6,8,8,6,4,3,3,3};
```


Contando operaciones

Pensar en el peor y mejor caso.

```
void ordenar(int arr[], int n) {  
    for (int i = 0; i < n-1; i++) {  
        for (int j = 0; j < n-1; j++) {  
            if(arr[j] > arr[j+1]) {  
                // swap  
                int temp = arr[j];  
                arr[j] = arr[j+1];  
                arr[j+1] = temp;  
            }  
        }  
    }  
}
```

$$(n-1)*((n-1) * (1+1+1+1))$$

$$(n-1)*((n-1) * 4)$$

$$(n-1)*(4n - 4)$$

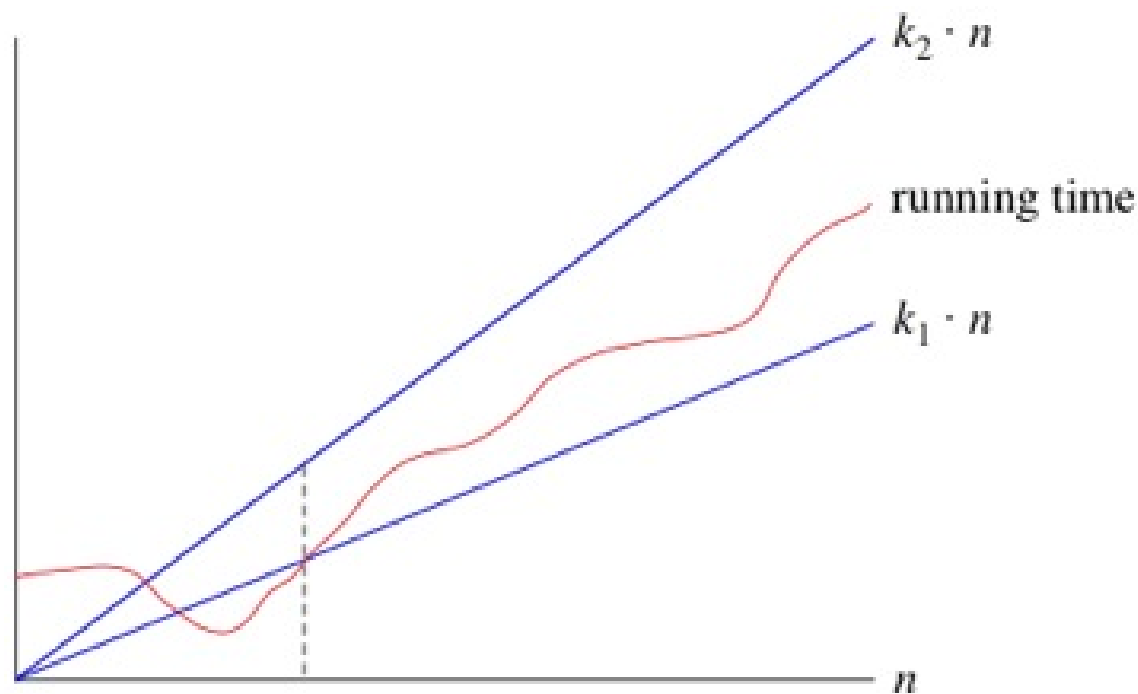
$$4n^2-4n-4n+4$$

$$4n^2-8n+4$$

Tasa de crecimiento: n^2

Notación asintótica - $\Theta(n)$

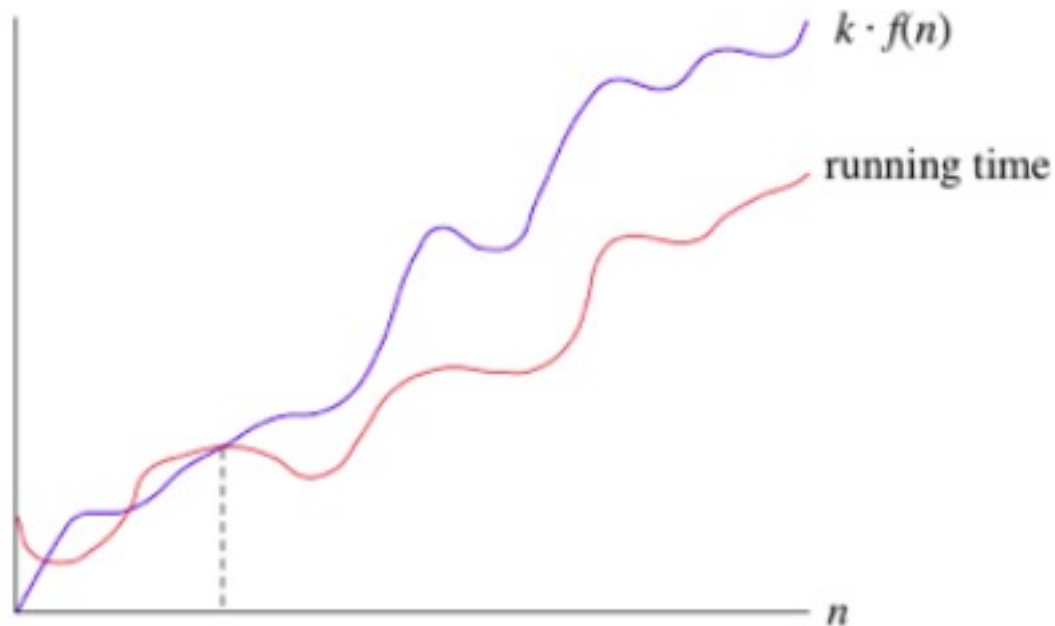
Cuando decimos que un tiempo de ejecución particular es de $\Theta(n)$, estamos diciendo que una vez que n sea suficientemente grande, el tiempo de ejecución será por lo menos $k_1 \cdot n$ y a lo más $k_2 \cdot n$ para algunas constantes k_1 y k_2 . Aquí está cómo pensar acerca de $\Theta(n)$:



$$\log_a n = \frac{\log_b n}{\log_b a}$$

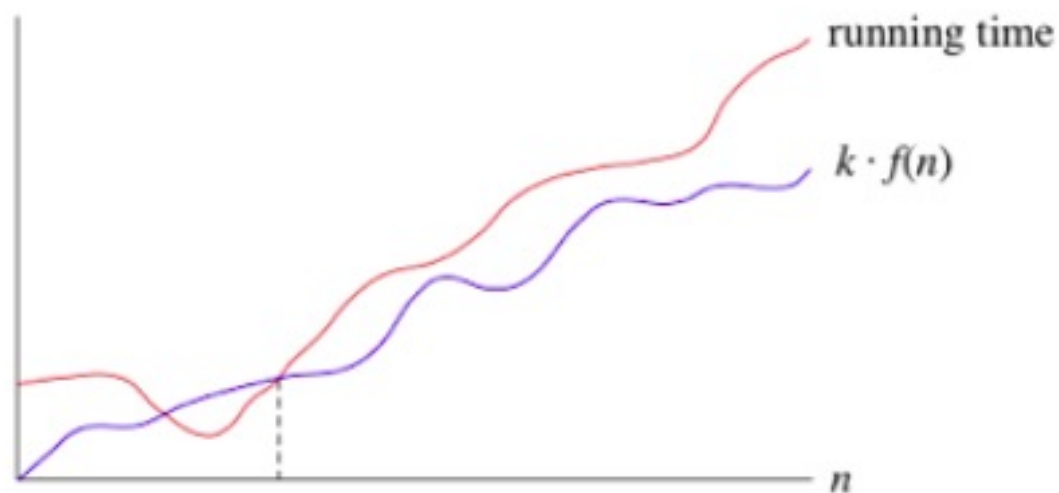
Notación asintótica - $O(n)$

Si un tiempo de ejecución es $O(f(n))$, entonces para n suficientemente grande, el tiempo de ejecución es a lo más $k \cdot f(n)$ para alguna constante k . Aquí está cómo pensar acerca de un tiempo de ejecución que es $O(f(n))$:



Notación asintótica - $\Omega(n)$

Si un tiempo de ejecución es $\Omega(f(n))$, entonces para una n suficientemente grande, el tiempo de ejecución es por lo menos $k \cdot f(n)$ para alguna constante k . Aquí está cómo pensar acerca de un tiempo de ejecución que es $\Omega(f(n))$:



Límite asintótico inferior