Recurrencias

mat-152

Recurrencias

- Una recurrencia es una ecuación o desigualdad que describe una función en términos de su valor en instancias más pequeñas.

- Para mergesort encontramos la recurrencia:

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1 \\ 2T(n/2) + \Theta(n) & \text{if } n > 1 \end{cases}$$

- Tres métodos principales de resolución:
    - ➡ Método de sustitución.
    - ➡ Método del árbol recursivo.
    - ➡ Método maestro.

Otros ejemplos de recurrencias

$T(n) = T(n-1) + n$ $\qquad \Theta(n^2)$

- Algoritmo recursivo que cicla en la entrada para eliminar un elemento.

$T(n) = T(n/2) + c$ $\qquad \Theta(\lg n)$

- Algoritmo recursivo que divide la entrada a la mitad en cada paso.

$T(n/2) + n$ $\qquad \Theta(n)$

- Algoritmo recursivo que divide la entrada a la mitad pero debe examinar cada elemento de la entrada.

$T(n) = 2T(n/2) + 1$ $\qquad \Theta(n)$

- Algoritmo que divide la entrada en 2 mitades y hace un trabajo constante.

Recurrencias

- Suposiciones:
  - ➡ Meta: obtener un estimado asintótico (en términos de la notación O,Ω o Θ.
  - ➡ Para valores pequeños de n, los valores de la recurrencia son constantes.

Método de sustitución

1. Adivinar la forma de la solución.
2. Verificar por inducción.
3. Resolver para las constantes.

Puede servir para encontrar tanto las cotas inferiores como las cotas superiores para una recurrencia.

Método de sustitución: ejemplo

Encontrar una cota superior para la recurrencia:

$$T(n) = 2T(\lfloor n/2 \rfloor) + n$$

1. Adivinamos una solución:

$$T(n) = O(n \lg n)$$

2. Probar por inducción que:

$$T(n) \leq cn \lg n$$

3. Para una elección apropiada de constante $c > 0$

Método de sustitución: ejemplo

Suponemos que esta cota superior se mantiene para $\lfloor n/2 \rfloor$, esto es que:

$$T(\lfloor n/2 \rfloor) \leq c\lfloor n/2 \rfloor \lg(\lfloor n/2 \rfloor)$$

y sustituyendo en la recurrencia $T(n) = 2T(\lfloor n/2 \rfloor) + n$ tenemos:

$$\begin{aligned} T(n) &\leq 2(c\lfloor n/2 \rfloor \lg(\lfloor n/2 \rfloor)) + n \\ &\leq cn \lg(n/2) + n \\ &= cn \lg n - cn \lg 2 + n \\ &= cn \lg n - cn + n \\ &\leq cn \lg n \end{aligned}$$

siempre y cuando $c \geq 1$.

Método de sustitución: ejemplo

- Para probar por inducción necesitamos identificar casos base.
- Mostrar que podemos elegir la constante *c* de tal manera que la cota superior sea valida también para los casos base.

  Supongamos entonces que el caso base es $T(1) = 1$, y para $n = 1$

$$T(n) \leq cn \lg n$$

$$T(1) \leq c1 \lg 1$$

$$= 0$$

- Podemos usar entonces $n_0 = 2$ y como casos base de la inducción $n = 2$ y $n = 3$ que resultan en valores de $T(2) = 4$ y $T(3) = 5$ para la recurrencia.

Método de sustitución: ejemplo

- Se puede completar entonces la prueba inductiva que $T(n) \leq cn \lg n$ para alguna constante $c \geq 1$ eligiendo una *c* de manera que $T(2) \leq c2 \lg 2$ y que $T(3) \leq c3 \lg 3$.
- Cualquier $c \geq 2$ mantiene los casos base verdaderos.

Método de sustitución: conclusiones

- No existe una forma general de solución.
- Adivinar la solución requiere experiencia y creatividad.
- Se puede utilizar otro método para encontrar la solución y probar con el método de sustitución.
- Se pueden poner cotas superiores e inferiores más laxas para ir restringiendo la solución.

Método del árbol recursivo

- Convierte la recurrencia en un árbol cuyos nodos representan los costos incurridos en cada nivel de la recursión.
- Cada nodo representa el costo de un solo subproblema en algún lugar de las llamadas recursivas.
- Estos costos se suman por nivel para luego determinar el costo total de la recursión.
- Este método es particularmente útil para analizar algoritmos de tipo *divide-and-conquer*.
- Sirven en su mayoría para generar buenas hipótesis para verificar por el método de sustitución.

Método del árbol recursivo: ejemplo

Encontrar las cotas para la recurrencia:

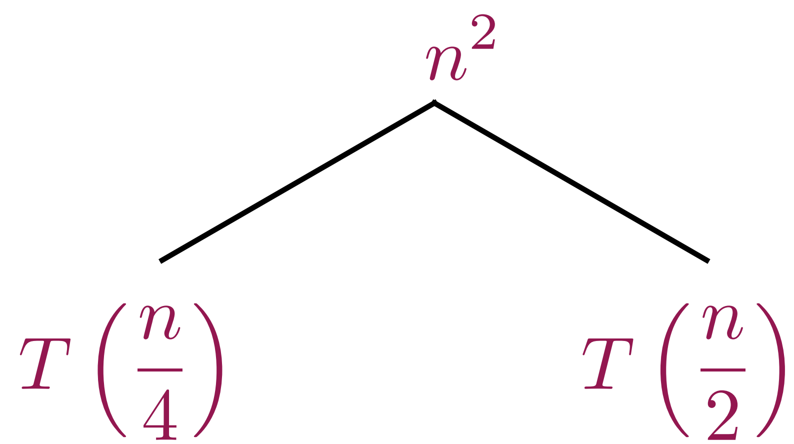
$$T(n) = T(n/4) + T(n/2) + n^2$$

$$T(n)$$

Método del árbol recursivo: ejemplo

Encontrar las cotas para la recurrencia:

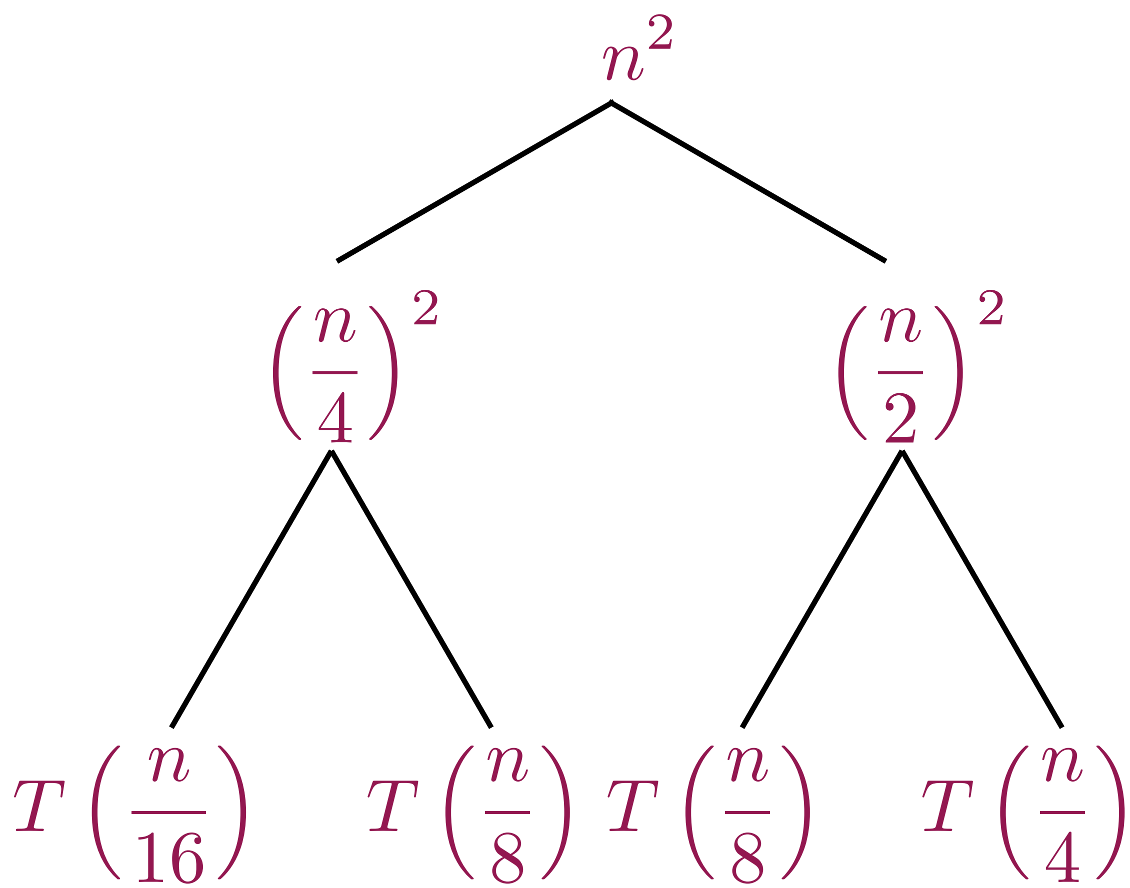
$$T(n) = T(n/4) + T(n/2) + n^2$$



Método del árbol recursivo: ejemplo

Encontrar las cotas para la recurrencia:

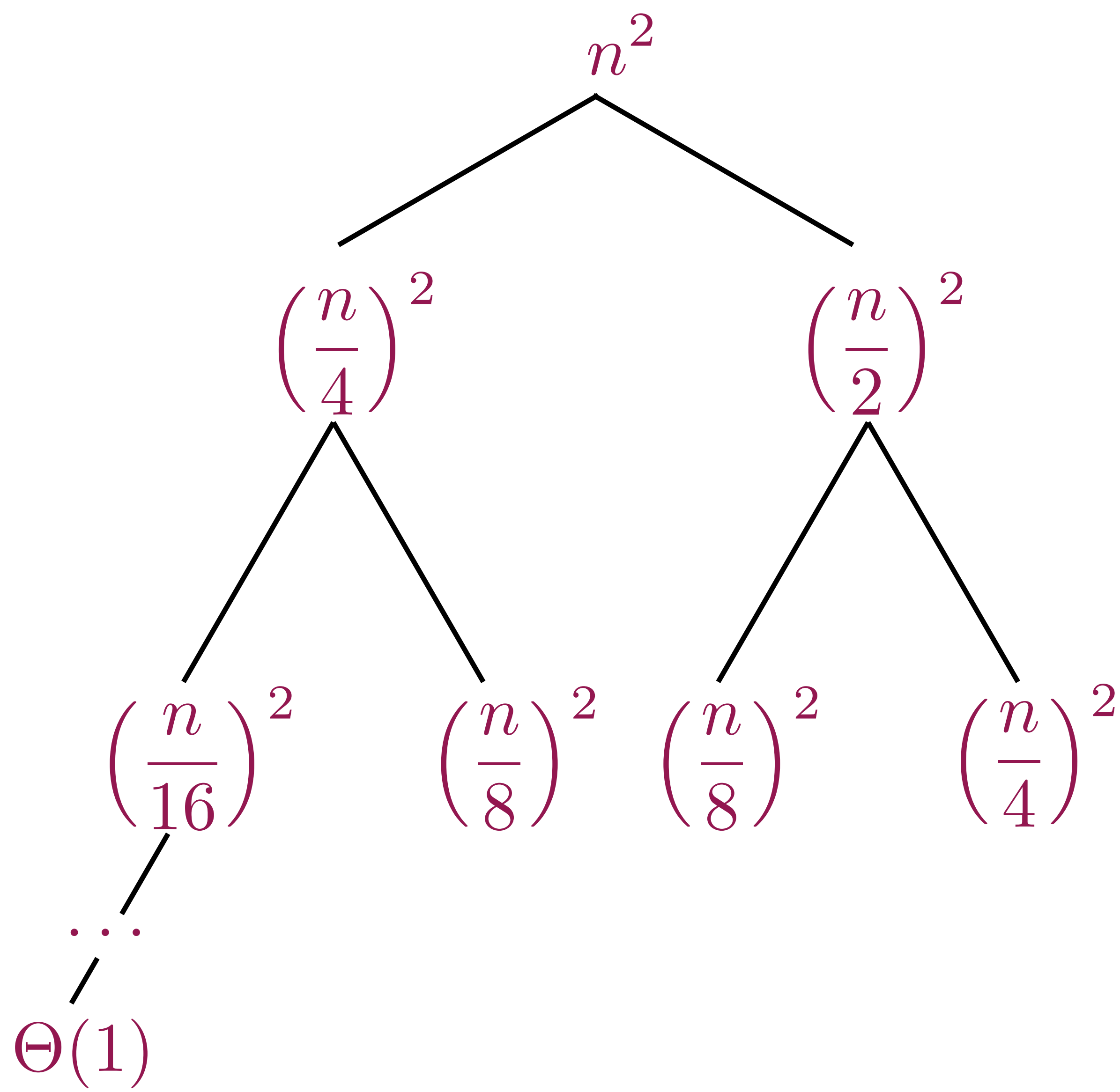
$$T(n) = T(n/4) + T(n/2) + n^2$$



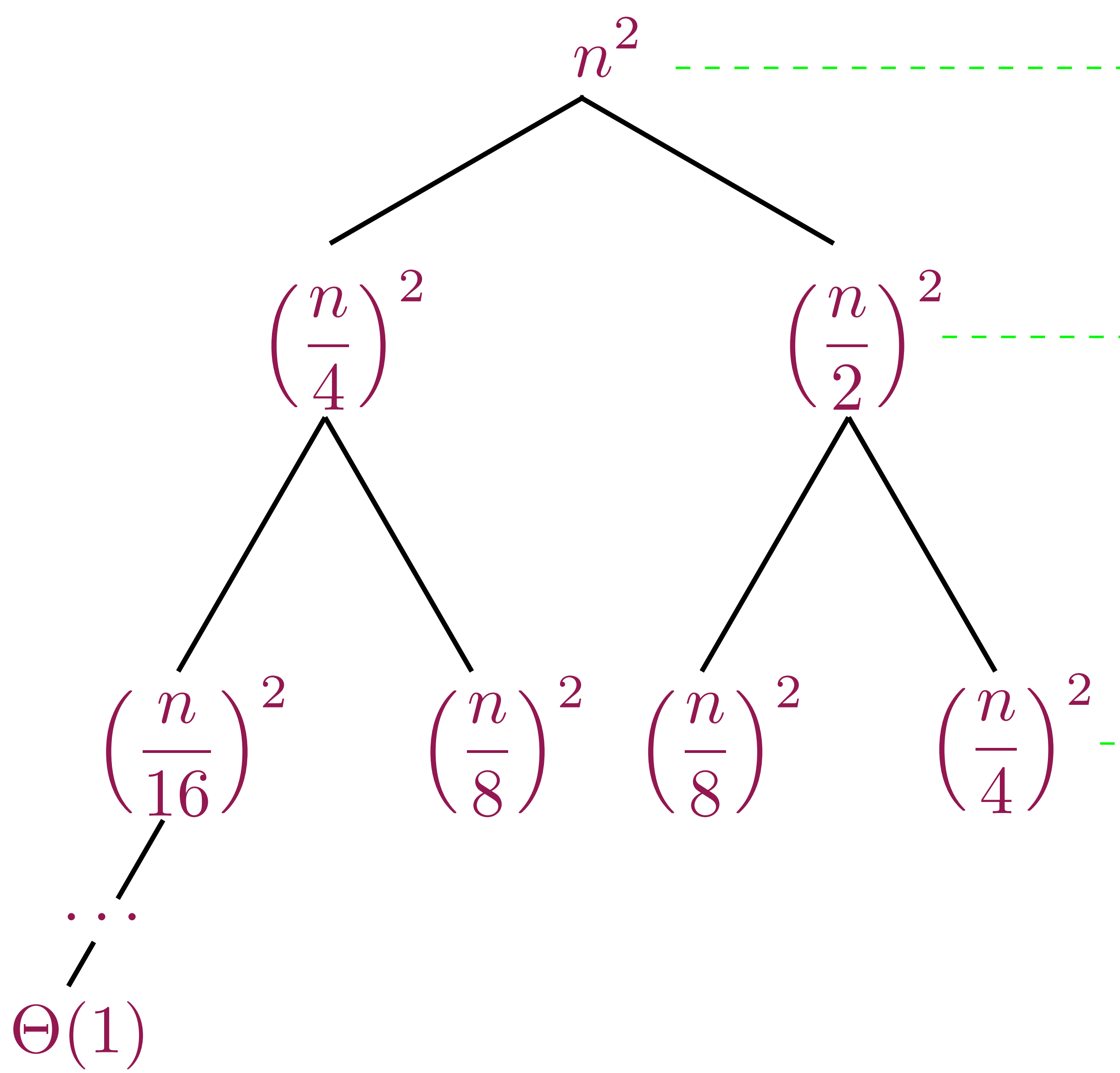
Método del árbol recursivo: ejemplo

Encontrar las cotas para la recurrencia:

$$T(n) = T(n/4) + T(n/2) + n^2$$



Método del árbol recursivo: ejemplo



$$n^2$$

$$\frac{n^2}{16} + \frac{n^2}{4} = \frac{5}{16}n^2$$

$$\frac{n^2}{256} + \frac{n^2}{64} + \frac{n^2}{64} + \frac{n^2}{16} = \frac{25}{256}n^2$$

$$\vdots$$

$$= n^2 \left(1 + \frac{5}{16} + \left(\frac{5}{16}\right)^2 + \left(\frac{5}{16}\right)^3 + \dots\right)$$

Método del árbol recursivo (paréntesis)

Serie geométrica:

$$1 + x + x^2 + \ldots + x^n = \frac{1 - x^{n+1}}{1 - x} \text{ para } x \neq 1,$$

$$1 + x + x^2 + \ldots = \frac{1}{1 - x} \text{ para } |x| < 1.$$

Método del árbol recursivo: ejemplo

$$= n^2 \left(1 + \frac{5}{16} + \left(\frac{5}{16}\right)^2 + \left(\frac{5}{16}\right)^3 + \dots\right)$$

$$= \Theta(n^2)$$

- Serie geométrica!

Método maestro

- Proporciona una manera de resolver recurrencias de la forma:

$$T(n) = aT(n/b) + f(n)$$

  donde $a \geq 1$ y $b > 1$ son constantes y $f(n)$ es una función asintóticamente positiva.

- Esta recurrencia describe el tiempo de un algoritmo que ...
  - ➡ divide un problema de tamaño n en a subproblemas, cada uno de tamaño n/b, donde a y b son constantes positivas.
  - ➡ Los a subproblemas se resuelven recursivamente, cada uno en un tiempo $T(n/b)$.
  - ➡ El costo de dividir el problema y combinar los resultados de los subproblemas está descrito por la función $f(n)$.

Método maestro (paréntesis)

- Como precisión técnica, la recurrencia no está bien definida ya que n/b puede no ser entero.
- Reemplazar los *a* términos $T(n/b)$ con $T(\lfloor n/b \rfloor)$ o $T(\lceil n/b \rceil)$ no modifica el comportamiento asintótico.

Teorema maestro

- El método maestro depende del teorema siguiente:

➡ Sean $a \geq 1$ y $b > 1$ constantes.

➡ Sea $f(n)$ una función.

➡ Sea $T(n)$ una función definida en los enteros positivos por la recurrencia:

$$T(n) = aT(n/b) + f(n)$$

donde interpretamos que n/b significa ya sea $\lfloor n/b \rfloor$ o $\lceil n/b \rceil$. Entonces $T(n)$ puede estar acotada asintóticamente de la forma siguiente:

Teorema maestro: tres casos

1. Si $f(n) = O(n^{\log_b a-\epsilon})$ para alguna constante $\epsilon > 0$, entonces

$$T(n) = \Theta(n^{\log_b a})$$

2. Si $f(n) = \Theta(n^{\log_b a})$, entonces

$$T(n) = \Theta(n^{\log_b a} \lg n)$$

3. Si $f(n) = \Omega(n^{\log_b a+\epsilon})$ para alguna constante $\epsilon > 0$, y si $af(n/b) \leq cf(n)$ para alguna constante $c < 1$ y una n suficientemente grande, entonces

$$T(n) = \Theta(f(n))$$

■

Teorema maestro

- En los tres casos estamos comparando la función $f(n)$ con la función $n^{\log_b a}$
- Intuitivamente, la solución a la recurrencia estará determinada por la mayor de estas funciones.
- En 1. la función $n^{\log_b a}$ crece más rápido polinomialmente que $f(n)$ por un factor n^ϵ. La solución esta entonces dada por $n^{\log_b a}$.
- En 3. la función $f(n)$ crece polinomialmente más rápido que $n^{\log_b a}$ por un factor de n^ϵ . Además $f(n)$ satisface la condición de regularidad que dice que $af(n/b) \leq cf(n)$ para alguna constante $c > 1$.
- Los tres casos no cubren todas las posibilidades, el método no puede aplicarse si las funciones no crecen polinomialmente mas rápido o si la condición de regularidad no se cumple.

Teorema maestro: ejemplo 1

$$T(n) = 4T(n/2) + n$$

$$a = 4 \, , \, b = 2$$

$$\Rightarrow n^{\log_b a} = n^{\log_2 4} = n^2$$

$$f(n) = n$$

$f(n) = O(n^{2-\epsilon})$ para $\epsilon = 1$

Entonces, por el caso 1 del TM tenemos que: $4T(n/2) + n = \Theta(n^2)$.

Teorema maestro: ejemplo 2

$$T(n) = 3T(n/2) + n$$

$$a = 3,\ b = 2$$

$$\Rightarrow n^{\log_b a} = n^{\log_2 3}$$

$f(n) = n$ caso 1: $f(n) = O(n^{\log_b a - \epsilon})$

$n = O(n^{\log_2 3 - \epsilon})$ con $\epsilon = n^{\log_2 3} - 1 \approx 0.59$

Entonces, por el caso 1 del TM tenemos que: $T(n) = \Theta(n^{\log_2 3})$.

Teorema maestro: ejemplo 3

$$T(n) = 2T(n/2) + n$$

$$a = 2,\ b = 2$$

$$\Rightarrow n^{\log_b a} = n^{\log_2 2} = n$$

$$f(n) = n$$

caso 2: $f(n) = \Theta(n^{\log_b a})$

$$f(n) = \Theta(n)?$$

Entonces, por el caso 2 del TM tenemos que: $T(n) = \Theta(n \lg n)$.

Teorema maestro: ejemplo 5

$$T(n) = 4T(n/2) + n^2$$

$$a = 4,\ b = 2$$

$$\Rightarrow n^{\log_b a} = n^{\log_2 4} = n^2$$

$$f(n) = n^2$$

$$n^2 = \Theta(n^2)$$

Entonces, por el caso 2 del TM tenemos que: $T(n) = \Theta(n^2 \lg n)$.

Teorema maestro: ejemplo 6

$$T(n) = 4T(n/2) + n^3$$

$$a = 4,\ b = 2$$

$$\Rightarrow n^{\log_b a} = n^{\log_2 4} = n^2$$

$$f(n) = n^3$$

caso 3:

$$f(n) = \Omega(n^{\log_b a + \epsilon}) \text{ para } \epsilon > 0$$

$$\text{y } af(n/b) \leq cf(n) \text{ para } c < 1$$

$$n^3 = \Omega(n^{2+\epsilon}) \text{ para } \epsilon > 0?$$

$$n^3 = \Omega(n^3) \text{ para } \epsilon = 1$$

$$4\left(\frac{n}{2}\right)^3 \leq cn^3 \text{ para } c < 1?$$

$$\text{para } c = \frac{1}{2}$$

Entonces, por el caso 3 del TM tenemos que: $T(n) = \Theta(n^3)$.

Teorema maestro: ejemplo 7

$$T(n) = 2T(n/2) + n \log_2 n$$

$a = 2,\ b = 2$

$\Rightarrow \log_b a = 1$

$f(n) = n \log_2 n$

$f(n) \neq O(n^{\log_2 a - \epsilon})$ para ningun $\epsilon > 0$. Entonces el caso 1 de TM no se aplica.

$f(n) \neq \Theta(n^{\log_2 a})$, por lo que el caso 2 tampoco aplica.

$f(n) \neq \Omega(n^{1+\epsilon})$ para ningun $\epsilon > 0$ (ya que $\log_2 n = o(n^{\epsilon})$ para todo $\epsilon > 0$).

El caso 3 tampoco aplica.

Teorema maestro

- Interesados en la prueba del teorema maestro, leer el capítulo 4, sección 4.4 del libro de CLRS.