Fundamentos de análisis y diseño de algoritmos

Ecuaciones de recurrencia

Recurrencias

Método de iteración

Método maestro*

Método de sustitución

Recurrencias

Método de iteración

Expandir la recurrencia y expresarla como una suma de términos que dependen de n y de las condiciones iniciales

Recurrencias

$T(n) = n + 3T(n/4)$, $T(1) = \Theta(1)$ y n par

Expandir la recurrencia 2 veces

Recurrencias

$T(n) = n + 3\boxed{T(n/4)}$

$n + 3\ (\ n/4 + 3T(n/16))$

$n + 3\ (\ n/4 + 3(n/16\ \ + 3T(n/64)\ ))$

$n + 3^*n/4 + 3^2{}^*n/4^2\ \ + 3^3T(n/4^3)$

$T(1)$

$\boxed{T\left(\frac{n}{4}\right) = \frac{n}{4} + 3T\left(\frac{n}{4^2}\right)}$

$T\left(\frac{n}{4^2}\right) = \frac{n}{4^2} + 3T\left(\frac{n}{4^3}\right)$

Recurrencias

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

$n + 3\,(\,n/4 + 3T(n/16))$

$n + 3\,(\,n/4 + 3(n/16\ \ + 3T(n/64)\,))$

$n + 3*n/4 + 3^2*n/4^2\ \ + 3^3T(n/4^3)$

¿Cuándo se detienen las iteraciones?

Recurrencias

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

$n + 3\ (\ n/4 + 3T(n/16))$

$n + 3\ (\ n/4 + 3(n/16\ \ + 3T(n/64)\ ))$

$n + 3*n/4 + 3^2*n/4^2\ \ + 3^3T(n/4^3)$

¿Cuándo se detienen las iteraciones?

Cuando se llega a $T(1)$

Recurrencias

$3^i T\left(\frac{n}{4^i}\right)$

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

$n + 3\ (\ n/4 + 3T(n/16))$

$n + 3\ (\ n/4 + 3(n/16\ + 3T(n/64)\ ))$

$\hookrightarrow\ n + 3*n/4 + 3^2*n/4^2 + 3^3T(n/4^3)$

$n + 3\frac{n}{4} + n\left(\frac{3}{4}\right)^2 + n\left(\frac{3}{4}\right)^3 + n\left(\frac{3}{4}\right)^4 + \dots + n\left(\frac{3}{4}\right)^{i-1} + 3^i T\left(\frac{n}{4^i}\right)$

¿Cuándo se detienen las iteraciones?

Cuando se llega a T(1), esto es, cuando $(n/4^i)=1$

$4^i = n \qquad i = \log_4(n)$

$\left(\frac{3}{4}\right)^0 n + \left(\frac{3}{4}\right)n + \left(\frac{3}{4}\right)^2 n + \left(\frac{3}{4}\right)^3 n + \dots + \left(\frac{3}{4}\right)^{\log_4(n)-1} n + 3^{\log_4(n)} T(1)$

Recurrencias

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

$n + 3 (n/4 + 3T(n/16))$

$n + 3 (n/4 + 3(n/16 + 3T(n/64)))$

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

$n + 3*n/4 + 3^2*n/4^2 + 3^3n/4^3 + ... + 3^{\log 4n}T(1)$

¿Cuándo se detienen las iteraciones?

Cuando se llega a $T(1)$, esto es, cuando $(n/4^i)=1$

Recurrencias

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

$n + 3\,(\,n/4 + 3T(n/16))$

$n + 3\,(\,n/4 + 3(n/16\ + 3T(n/64)\,))$

$n + 3{*}n/4 + 3^2{*}n/4^2\ + 3^3T(n/4^3)$

$n + 3{*}n/4 + 3^2{*}n/4^2 + 3^3n/4^3 + \ldots + 3^{\log 4n}T(1)$

Después de iterar, se debe tratar de expresar como una sumatoria con forma cerrada conocida

Recurrencias

$T(n) = n + 3T(n/4\rfloor)$

$n + 3\ (\ n/4\rfloor + 3T(n/16\rfloor))$

$n + 3\ (\ n/4\rfloor + 3(n/16\rfloor\ + 3T(n/64\rfloor)\ ))$

$n + 3{*}n/4\rfloor + 3^2{*}n/4^2\rfloor\ + 3^3(n/4^3\rfloor) + \ldots + 3^{\log 4n}\Theta(1)$

$\leq\ n + 3n/4 + 3^2n/4^2 + 3^3n/4^3 + \ldots + 3^{\log 4n}\Theta(1)$

Recurrencias

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

$n + 3\,(\,n/4 + 3T(n/16))$

$n + 3\,(\,n/4 + 3(n/16 \;+ 3T(n/64)\,))$

$n + 3{*}n/4 + 3^2{*}n/4^2 \;+ 3^3(n/4^3) + \ldots + 3^{\log 4n}\Theta(1)$

$\leq\; n + 3n/4 + 3^2n/4^2 + 3^3n/4^3 + \ldots + 3^{\log 4n}\Theta(1)$

$$= \left(\sum_{i=0}^{\log_4 n}\left(\frac{3}{4}\right)^i n\right) + 3^{\log_4 n}\Theta(1) \qquad n^{\log_4\left(\frac{3}{4}\right)}$$

$$= n\left(\frac{(3/4)^{(\log_4 n)} - 1}{(3/4) - 1}\right) + n^{\log_4 3} = n * 4(1 - (3/4)^{(\log_4 n)}) + \Theta(n^{\log_4 3})$$

$= O(n)$

Recurrencias

Resuelva por el método de iteración

$T(n) = 2\boxed{T(n/2)} + 1,\ T(1) = \Theta(1)$

$$T(n) = 2\left(2\boxed{T\left(\frac{n}{2^2}\right)} + 1\right) + 1$$

$$T(n) = 2\left(2\left(2\boxed{T\left(\frac{n}{2^3}\right)} + 1\right) + 1\right) + 1$$

$$T(n) = 2\left(2\left(2\left(2T\left(\frac{n}{2^4}\right) + 1\right) + 1\right) + 1\right) + 1$$

$$T(n) = 2^4\, T\left(\frac{n}{2^4}\right) + 2^3 + 2^2 + 2 + 2^0$$

$$T(n) = 2^i\, T\left(\frac{n}{2^i}\right) + \underline{2^{i-1} + 2^{i-2} + \dots + 2^2 + 2^1 + 2^0}$$

$T(1)$ $\quad 1 = \frac{n}{2^i} \quad i = \log_2(n)$

$$a^{\log_b(c)} = c^{\log_b(a)}$$

$$2^{\log_2(n)}\, T(1) + \sum_{i=0}^{\log_2(n)-1} 2^i$$

$$n^{\log_2(2)} \times C + \frac{2^{\log_2(n)-\cancel{1}+\cancel{1}} - 1}{2-1}$$

$$Cn + n - 1 = O(n)$$

Recurrencias

Resuelva por el método de iteración

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

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

$$T(n) = 2T\left(\frac{n}{2}\right) + 1$$

$$T(n) = 2\left(2T\left(\frac{n}{2^2}\right) + 1\right) + 1$$

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

$$T(n) = 2\left(2\left(2\left(2T\left(\frac{n}{2^4}\right) + 1\right) + 1\right) + 1\right) + 1$$

$$T(n) = 2^4 T\left(\frac{n}{2^4}\right) + 2^3 + 2^2 + 2^1 + 2^0$$

$$T(n) = 2^i T\left(\frac{n}{2^i}\right) + 2^{i-1} + 2^{i-2} + \ldots + 2^1 + 2^0$$

$$T(1) \qquad \frac{n}{2^i} = 1 \qquad i = \log_2(n)$$

$$T(n) = 2^{\log_2(n)} T(1) + 2^{\log_2(n)-1} + 2^{\log_2(n)-2} + \dots + 2^1 + 2^0$$

$$\boxed{\sum_{r=0}^{n} ar^i = \frac{ar^{n+1} - a}{r-1}}$$

$$T(n) = n\,T(1) + \sum_{i=0}^{\log_2(n)-1} 2^i$$

$$T(n) = n\,\Theta(1) + \frac{2^{\log_2(n)-1+1} - 1}{2-1}$$

$$T(n) = n\,\Theta(1) + n - 1 \longrightarrow \Theta(1)$$

$$T(n) = 2T\left(\frac{n}{2}\right) + n$$

$$2\left(2T\left(\frac{n}{2^2}\right) + \frac{n}{2}\right) + n$$

$$2\left(2\left(2T\left(\frac{n}{2^3}\right) + \frac{n}{2^2}\right) + \frac{n}{2}\right) + n$$

$$2^3T\left(\frac{n}{2^3}\right) + n + n + n$$

$$2^3T\left(\frac{n}{2^3}\right) + 3n$$

$$2^2T\left(\frac{n}{2^2}\right) + n + n$$

$$2^i T\left(\frac{n}{2^i}\right) + in$$

$$\frac{n}{2^i} = 1 \qquad i = \log_2(n)$$

$$2^{\log_2(n)} T(1) + \log_2(n)\, n$$

$$n\,\Theta(1) + n\log_2(n) = O(n\log(n))$$

$$T(n) = 2\boxed{T\left(\frac{n}{2}\right)} + n^2 \qquad \Theta(1) = T(1)$$

1) $$T(n) = 2\boxed{\left(2T\left(\frac{n}{2^2}\right) + \left(\frac{n}{2}\right)^2\right)} + n^2$$

$$T(n) = 2^2\boxed{T\left(\frac{n}{2^2}\right)} + 2\left(\frac{n}{2}\right)^2 + n^2$$

$$T(n) = 2^2\left(2T\left(\frac{n}{2^3}\right) + \left(\frac{n}{2^2}\right)^2\right) + 2\left(\frac{n}{2}\right)^2 + n^2$$

$$T(n) = 2^3T\left(\frac{n}{2^3}\right) + 2^2\left(\frac{n}{2^2}\right)^2 + 2\left(\frac{n}{2}\right)^2 + n^2$$

$$T(n) = 2^iT\left(\frac{n}{2^i}\right) + 2^{i-1}\left(\frac{n}{2^{i-1}}\right)^2 + 2^{i-2}\left(\frac{n}{2^{i-2}}\right)^2 + \dots + 2\left(\frac{n}{2}\right)^2 + 2^0\left(\frac{n}{2^0}\right)^2$$

$$i = \log_2(n)$$

$$T(n) = nT(1) + \sum_{i=0}^{\log_2(n)-1} 2^i \left(\frac{n}{2^i}\right)^2$$

$$\frac{\cancel{2^i}\, n^2}{(2^i)^{\cancel{2}}}$$

$$T(n) = n\Theta(1) + \sum_{i=0}^{\log_2(n)-1} n^2 \left(\frac{1}{2^i}\right)$$

$$T(n) = n\Theta(1) + n^2 \sum_{i=0}^{\log_2(n)-1} \underbrace{\left(\frac{1}{2}\right)}_{r}^{i}$$

$$T(n) = n\,\Theta(1) + n^2\left(\frac{\left(\frac{1}{2}\right)^{\log_2(n)} - 1}{\frac{1}{2} - 1}\right)$$

$$T(n) = n\,\Theta(1) + n^2\left(\frac{n^{\log_2(0.5)} - 1}{-\frac{1}{2}}\right)$$

$$T(n) = n\,\Theta(1) + n^2\left(-2\left(n^{-1} - 1\right)\right)$$

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

$$\Theta(n^2)$$

Recurrencias

Resuelva por el método de iteración

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

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

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

$$T(n) = 4T\left(\frac{n}{3}\right) + n + 1 \qquad T(1) = 1$$

Recurrencias

Resuelva por el método de iteración

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

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

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

Demuestre que $T(n) = T(n/2\rfloor) + n$, es $\Omega(n\log n)$

Recurrencias

Iteración con árboles de recursión

$T(1) = 1$

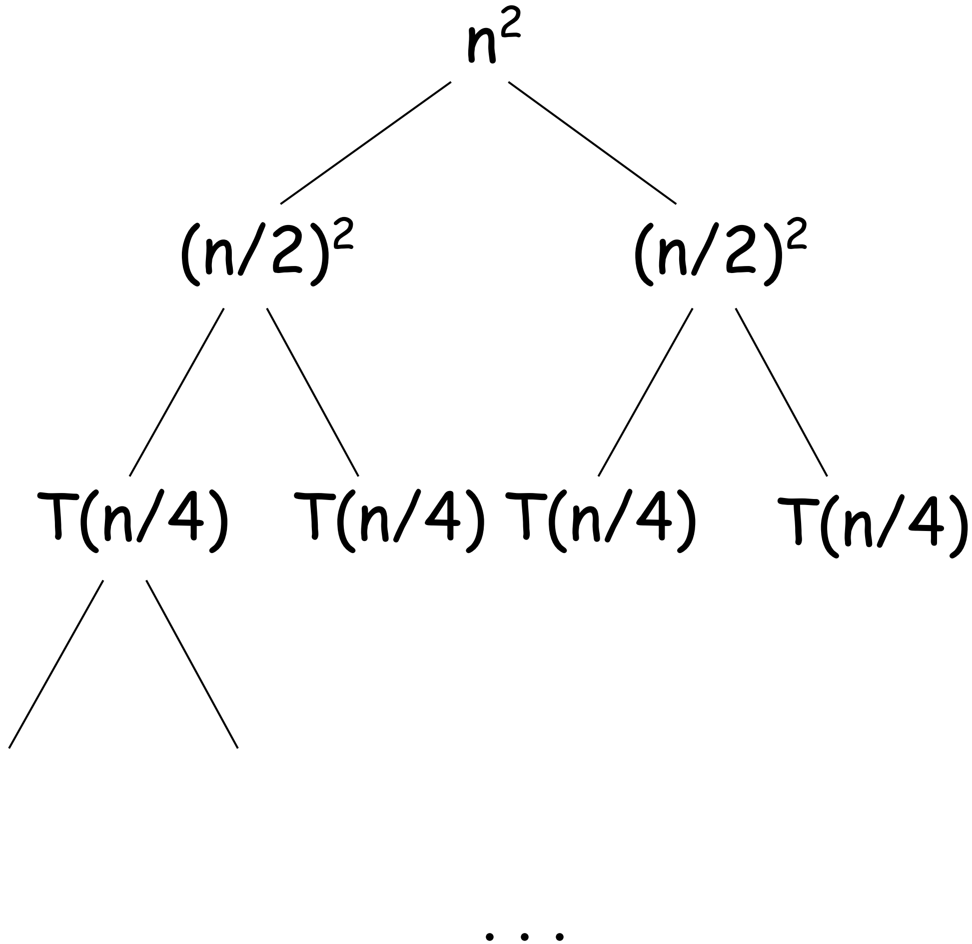
$T(n) = 2T(n/2) + n^2$

Recurrencias

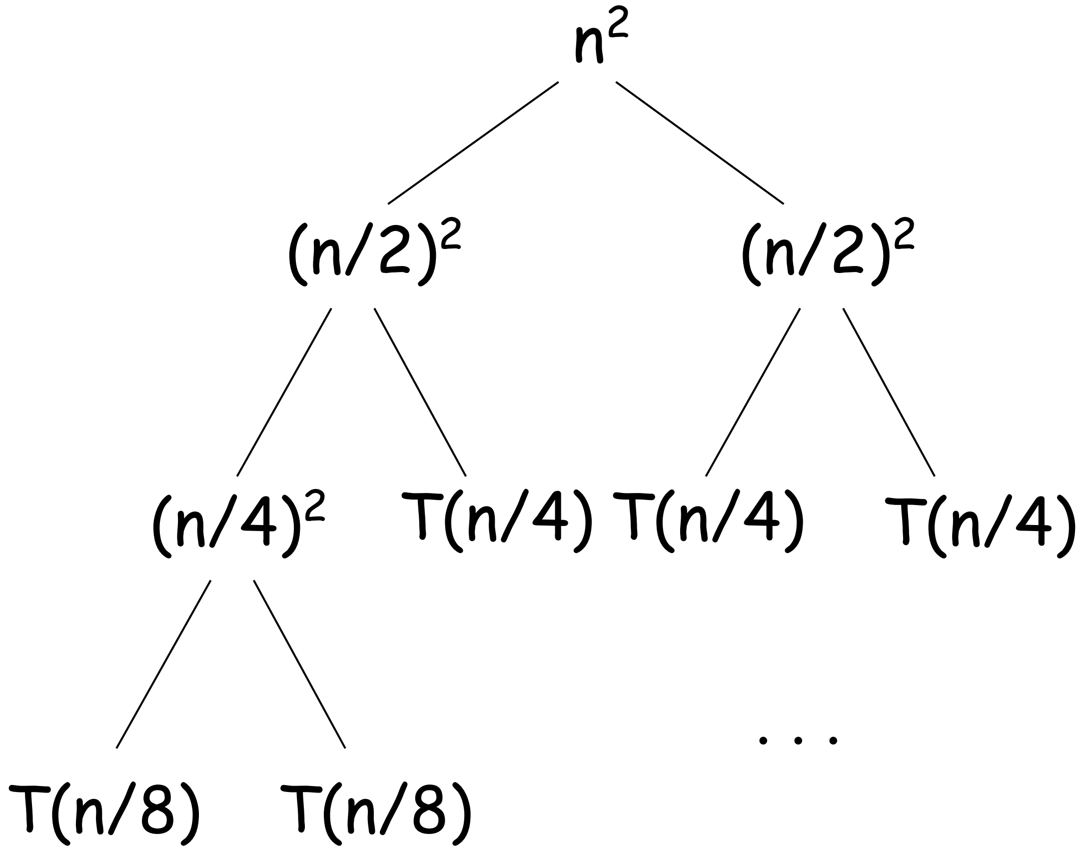
$$n^2$$

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

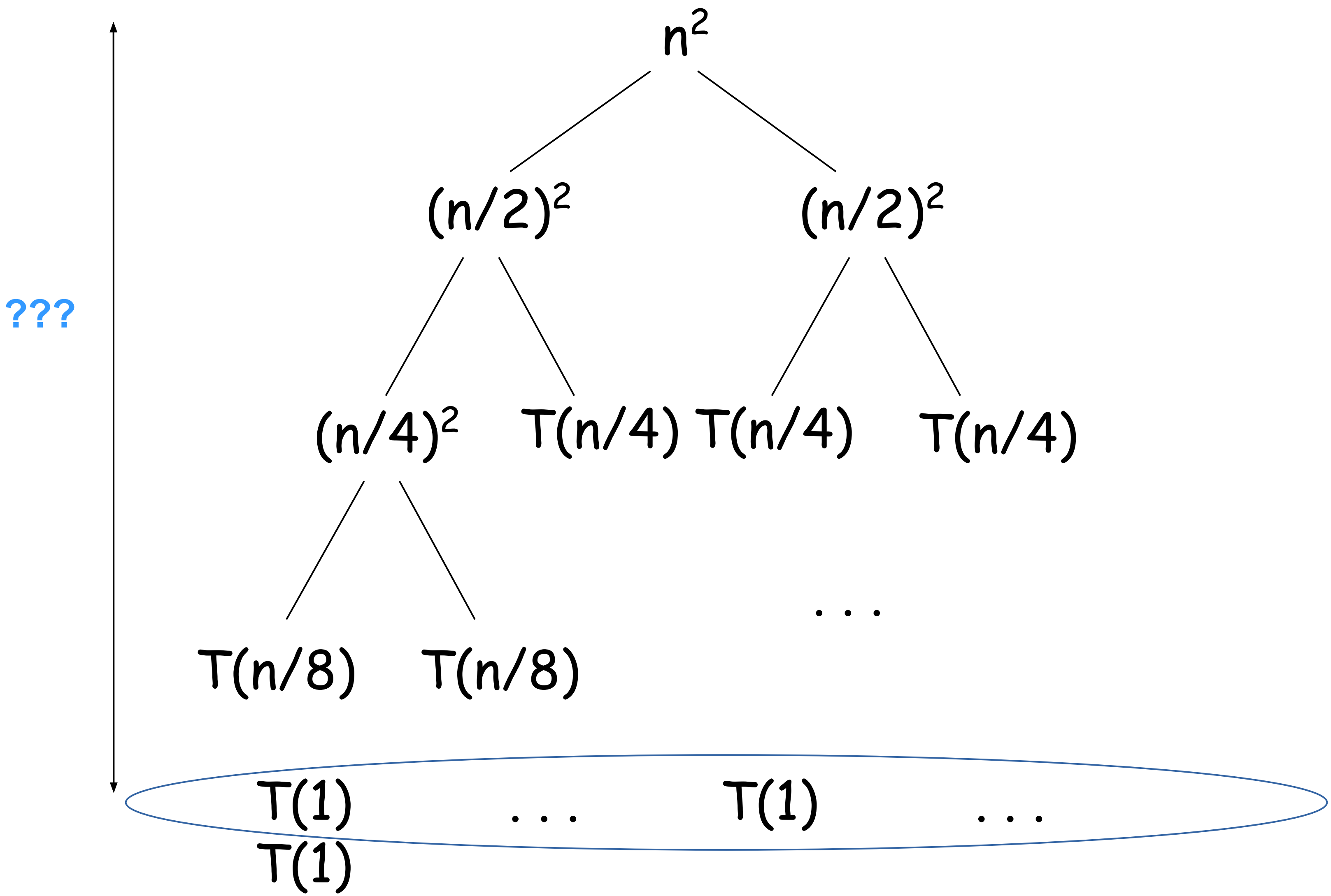
Recurrencias



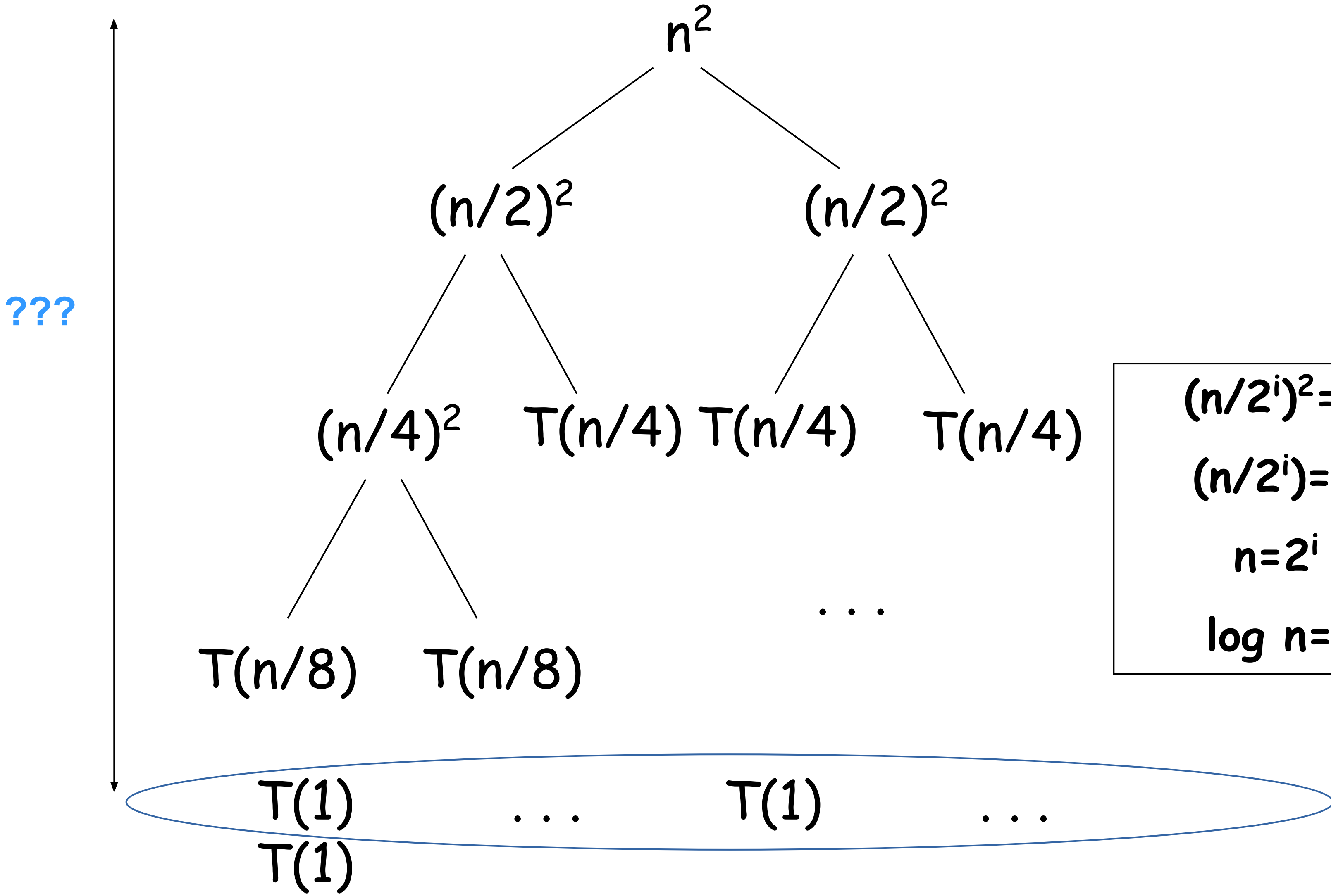
Recurrencias



Recurrencias



Recurrencias



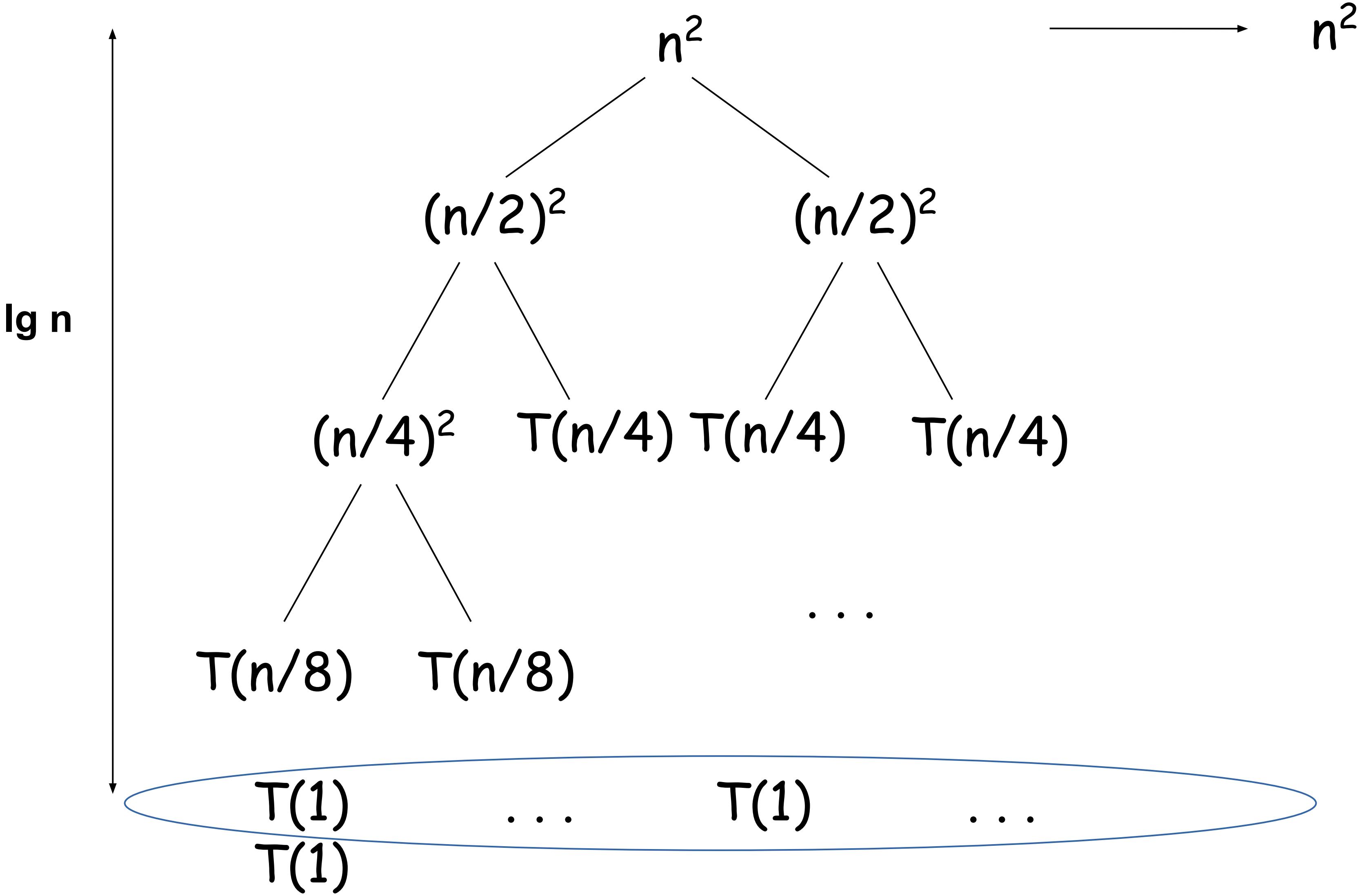
$$(n/2^i)^2=1$$

$$(n/2^i)=1$$

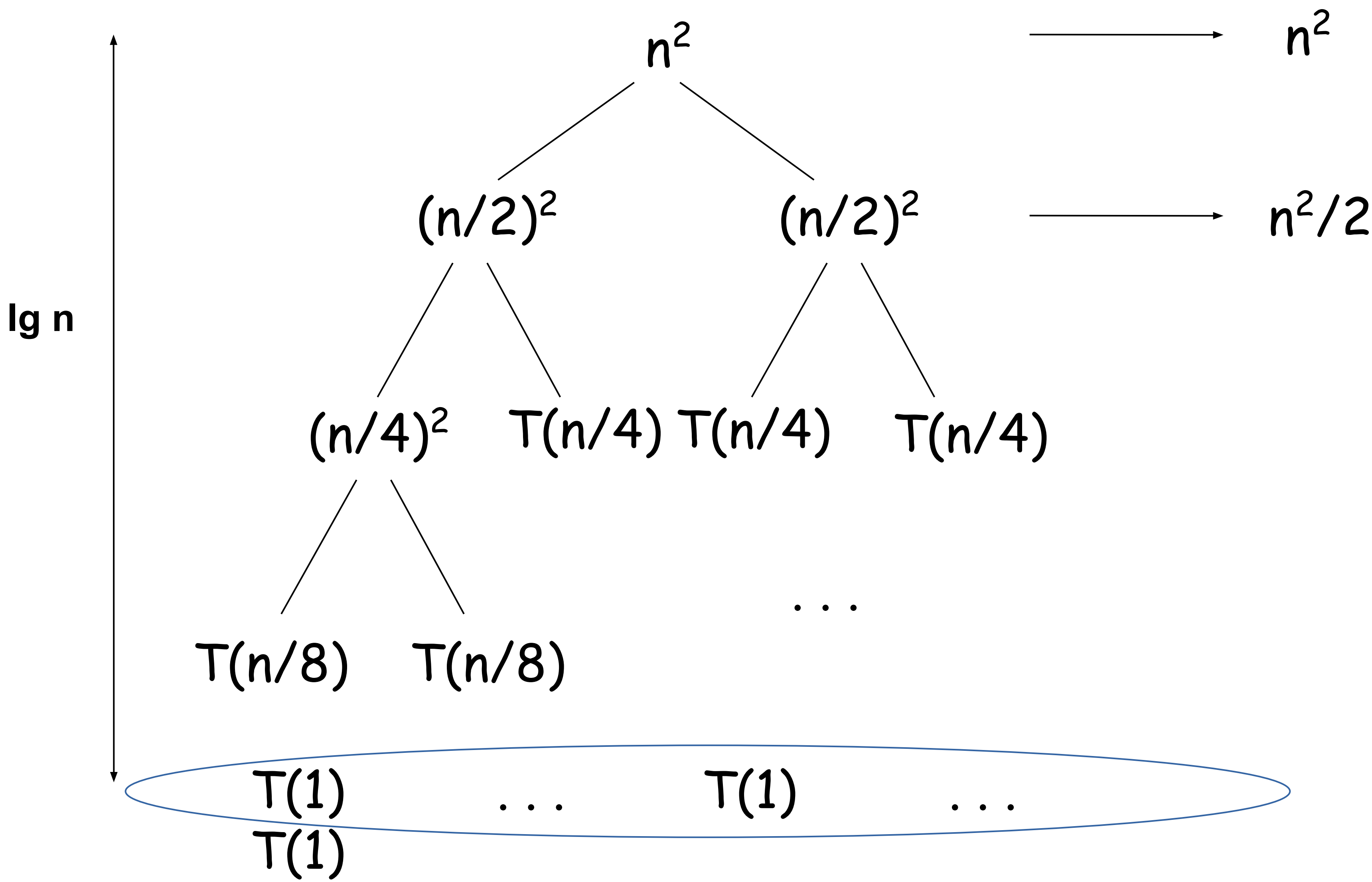
$$n=2^i$$

$$\log n=i$$

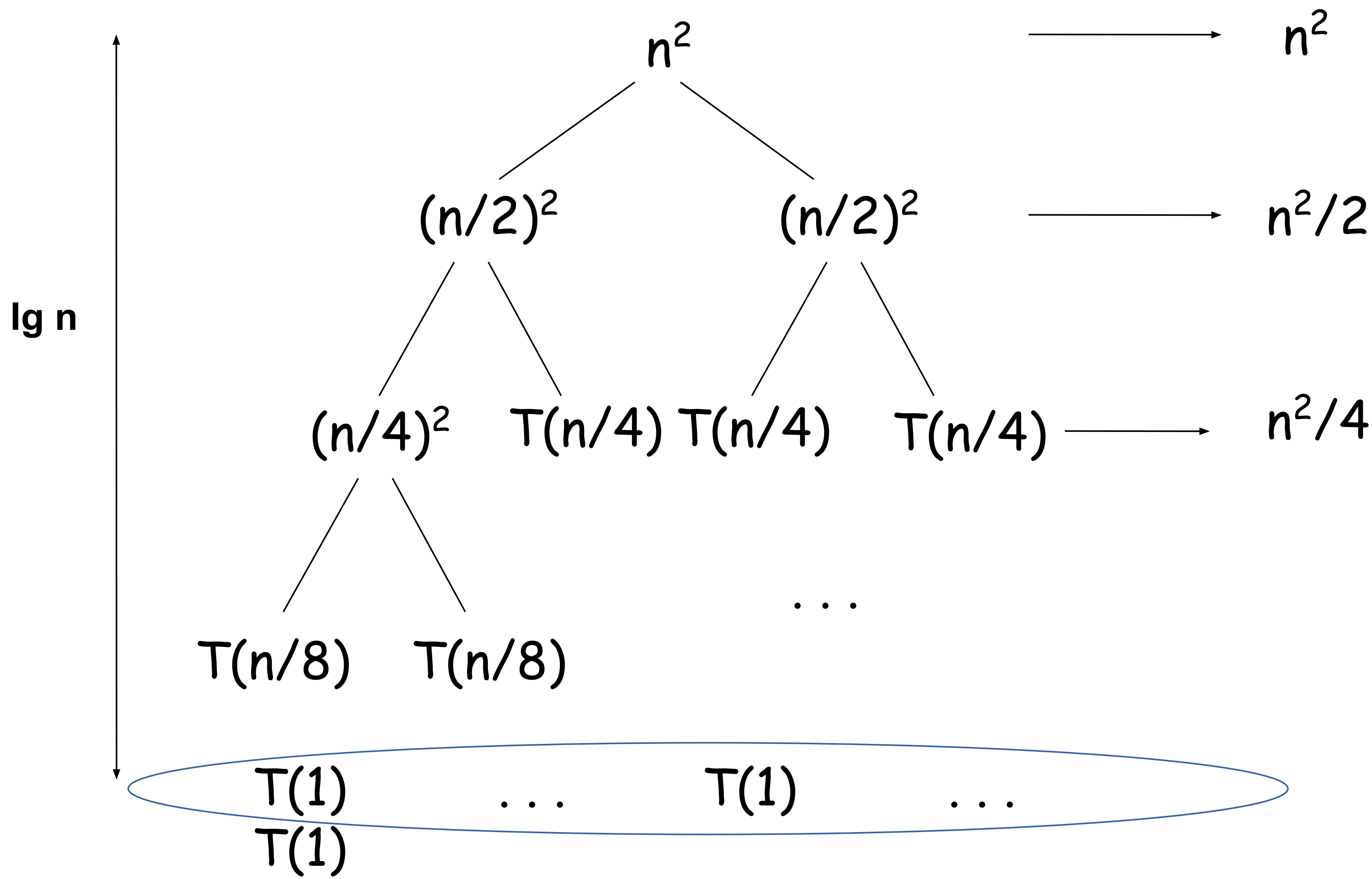
Recurrencias



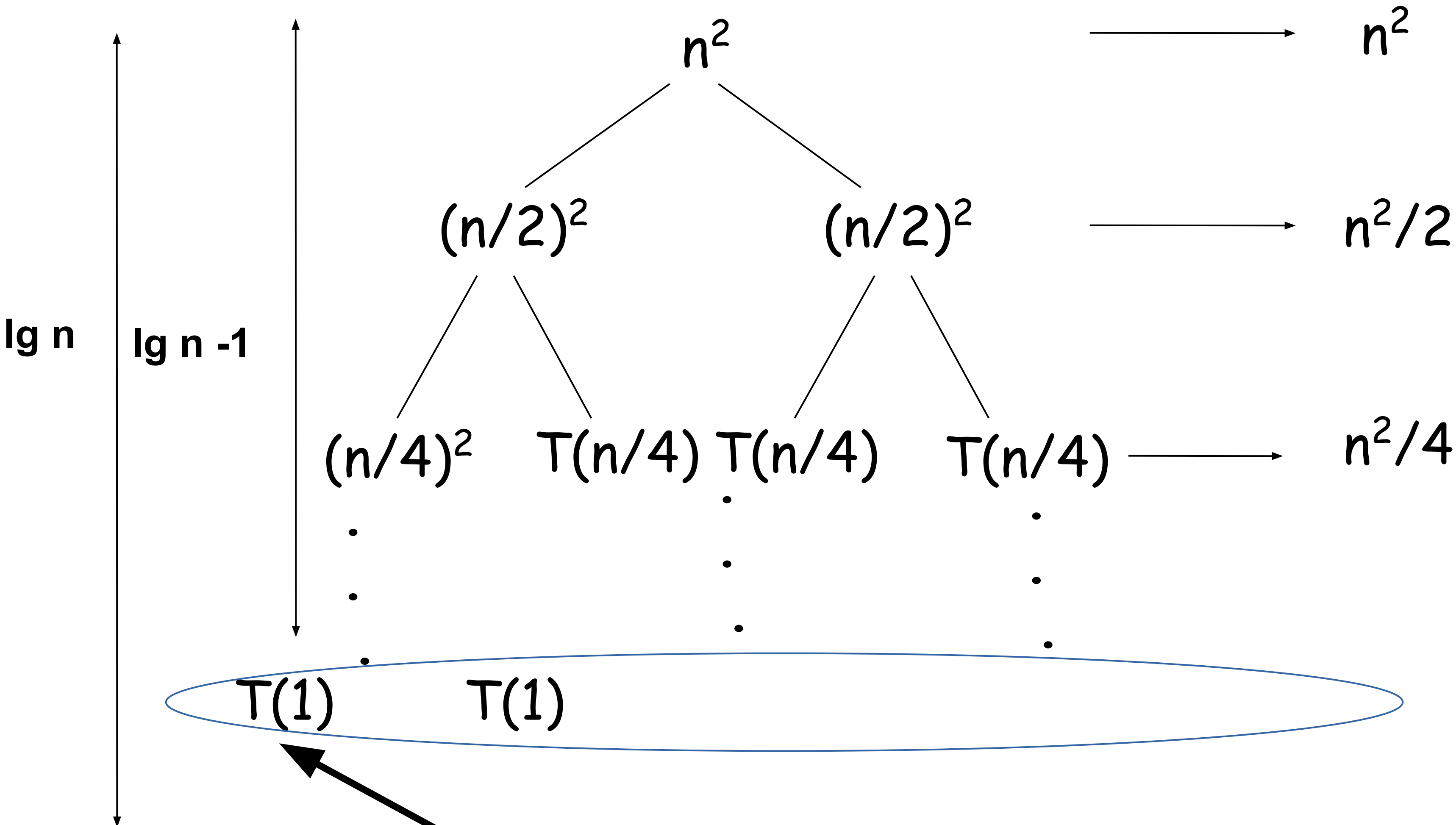
Recurrencias



Recurrencias

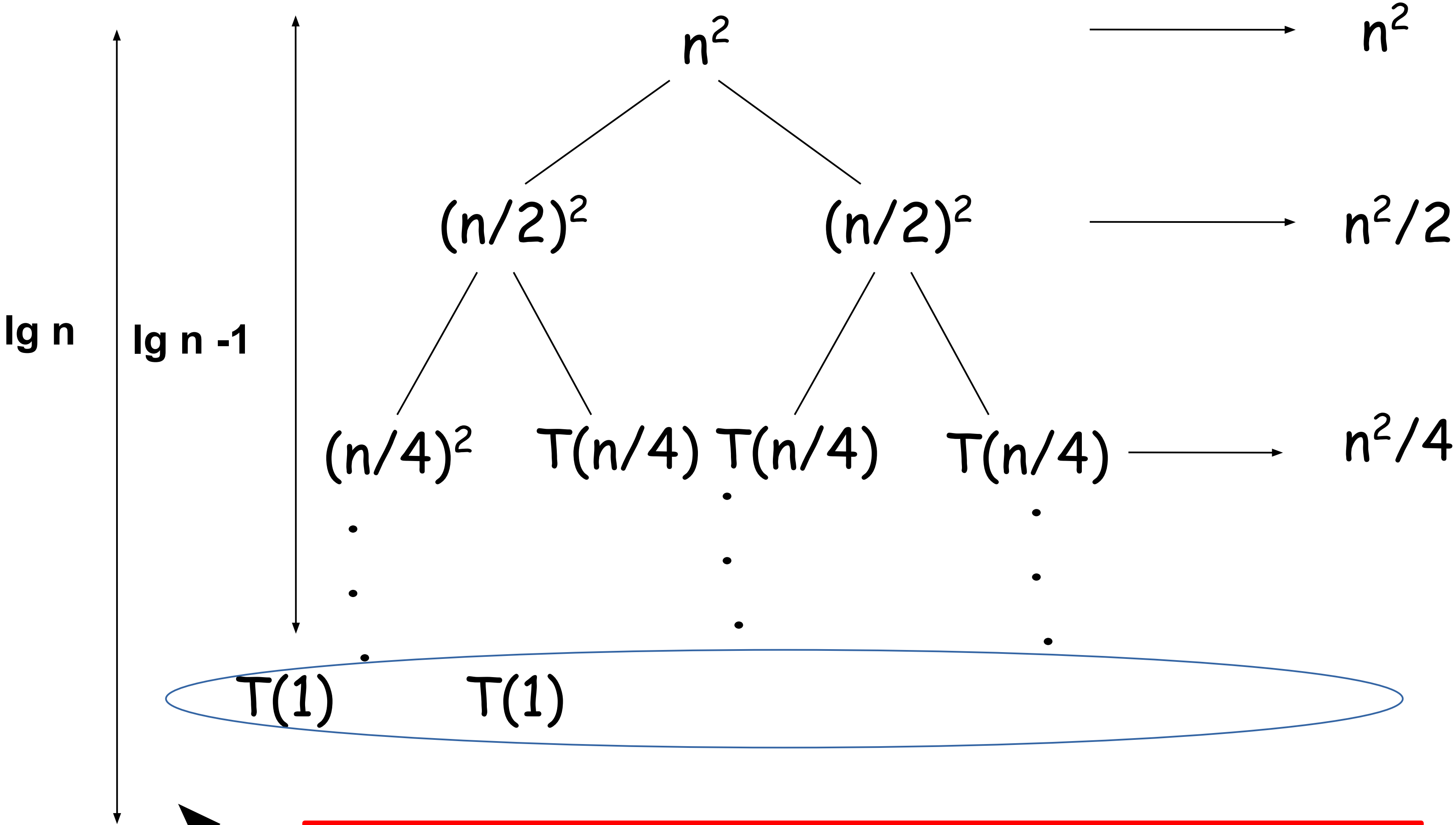


Recurrencias



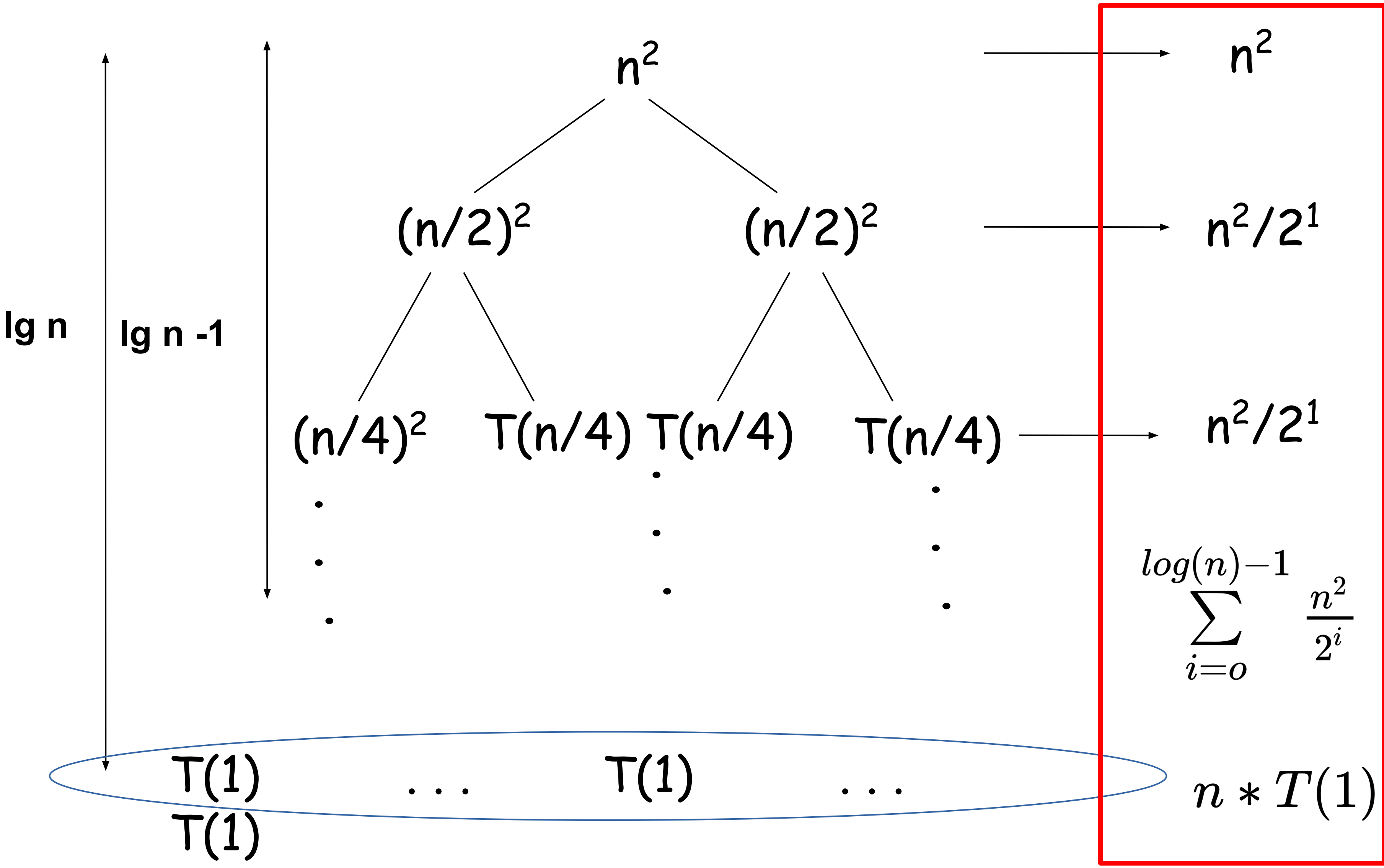
¿Cuantas hojas T(1) tenemos?

Recurrencias



Si recuerda en un árbol m-ario se tienen máximo m^h. En este caso al ser arbol binario m=2, tenemos $2^{\log(n)}$ hojas. Por lo tanto se

Recurrencias



Recurrencias

$$T(n) = n * T(1) + \sum_{i=o}^{log(n)-1} \frac{n^2}{2^i}$$

$$T(n) = n * c + n^2 \frac{0.5^{log(n)} - 1}{0.5 - 1}$$

$$T(n) = n * c + n^2 \frac{n^{log(0.5)} - 1}{-0.5}$$

$$T(n) = n * c + n^2 \frac{n^{-1} - 1}{-0.5}$$

$$T(n) = n * c - \frac{n}{0.5} + \frac{n^2}{0.5} = O(n^2)$$

Recurrencias

Resuelva construyendo el árbol

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

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

Recurrencias

Resuelva la recurrencia $T(n) = T(n/3) + T(2n/3) + n$

Indique una cota superior y una inferior

Recurrencias

Método maestro

Permite resolver recurrencias de la forma:

$$T(n) = aT(n/b) + f(n), \text{ donde } a \geq 1,\ b > 1$$

Recurrencias

Dado $T(n) = aT(n/b) + f(n)$, donde $a \geq 1$, $b>1$, se puede acotar asintóticamente como sigue:

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

Si $f(n)=O(n^{\log_b a-\varepsilon})$ para algún $\varepsilon>0$

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

Si $f(n)=\Theta(n^{\log_b a})$ para algún $\varepsilon>0$

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

Si $f(n)=\Omega(n^{\log_b a+\varepsilon})$ para algún $\varepsilon>0$ si a*f(n/b) ≤c*f(n)

para algun c<1

Recurrencias

Dado T(n) = 9T(n/3) + n

$$n^{\log_3 9} = n^2 \quad \textbf{Vs} \quad f(n) = n$$

Es $f(n) = O(n^{\log_b a - \varepsilon})$?

Es $n = O(n^{2-\varepsilon})$?

Recurrencias

Dado T(n) = 9T(n/3) + n

$$n^{\log_3 9} = n^2 \quad \textbf{Vs} \quad f(n) = n$$

Es $f(n) = O(n^{\log_b a - \varepsilon})$?

Es $n = O(n^{2-\varepsilon})$?

Si $\varepsilon = 1$ se cumple que $n = O(n)$, por lo tanto, se cumple que:

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

Recurrencias

T(n) = T(2n/3) + 1

$$n^{\log_{3/2} 1} = n^0 = 1 \quad \textbf{Vs} \quad f(n) = 1$$

Es $f(n) = O(n^{\log_b a - \varepsilon})$?

Es $1 = O(n^{0-\varepsilon})$?

No existe $\varepsilon > 0$

Recurrencias

T(n) = T(2n/3) + 1

$$n^{\log_{3/2} 1} = n^0 = 1 \quad \text{vs} \quad f(n) = 1$$

Es $f(n) = \Theta(n^{\log_b a})$?

Es $1 = \Theta(1)$?

Si, por lo tanto, se cumple que:

$$T(n) = \Theta(1 * \lg n) = \Theta(\lg n)$$

Recurrencias

T(n) =3 T(n/4) + nlgn

$$n^{\log_4 3} = n^{0.793} \quad \text{vs} \quad f(n) = n\lg n$$

Es $f(n) = O(n^{\log_b a - \varepsilon})$?

Es $f(n) = \Theta(n^{\log_b a})$?

Es $f(n) = \Omega(n^{\log_b a + \varepsilon})$?

Si, y además, $af(n/b) \leq cf(n)$

$3(n/4)\lg(n/4) \leq cn\lg n$

$3(n/4)\lg n - 3(n/4)*2 \leq cn\lg n$

$(3/4)n\lg n \leq cn\lg n \rightarrow c=3/4$ y se concluye que $T(n) = \Theta(n\lg n)$

Recurrencias

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

Muestre que no se puede resolver por el método maestro

Recurrencias

Resuelva usando método del maestro

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

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

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

Recurrencias

Método de sustitución

Suponer la forma de la solución y probar por inducción matemática

Recurrencias

$T(n)=2T(\lfloor n/2 \rfloor)+n$, $T(1)=1$

Suponer que la solución es de la forma $T(n)=O(n\lg n)$

Probar que $T(n)\leq cn\lg n$.

Se supone que se cumple para $n/2$ y se prueba para n

Hipotesis inductiva: $T(n/2) \leq cn/2\lg (n/2)$

Recurrencias

$T(n)=2T(\lfloor n/2 \rfloor)+n$, $T(1)=1$

Probar que $T(n) \leq cn\lg n$.

Hipótesis inductiva: $T(n/2) \leq cn/2\lg(n/2)$

Paso inductivo:

$$T(n) \leq 2(cn/2\lg(n/2)) + n$$
$$\leq cn \lg(n/2) + n$$
$$= cn \lg(n) - cn + n, \text{ para } c \geq 1, \text{ haga } c=1$$
$$\leq cn \lg n$$

Recurrencias

$T(n)=2T(\lfloor n/2 \rfloor)+n$, $T(1)=1$

Probar que $T(n) \leq cn\lg n$.

Paso base: si $c=1$, probar que $T(1)=1$ se cumple

$T(1) \leq 1*1\lg 1$?

$1 \leq 0$?

No, se debe escoger otro valor para c

Recurrencias

$T(n)=2T(\lfloor n/2 \rfloor)+n$, $T(1)=1$

Probar que $T(n) \le cn\lg n$.

Paso base: si $c=2$, probar que $T(1)=1$ se cumple

$T(1) \le 2*1\lg 1$?

$1 \le 0$?

No, se puede variar k.

Para esto, se calcula $T(2)$ y se toma como valor inicial

Recurrencias

Probar que $T(n) \leq cn\lg n$.

$T(2)=2T(0)+2=4$

Paso base: si $c=1$, probar que $T(2)=4$ se cumple

$T(2) \leq 1*2\lg 2$?

$4 \leq 2$?

No, se puede variar c.

Recurrencias

Probar que $T(n) \leq cn\lg n$.

$T(2)=2T(0)+2=4$

Paso base: si c=3, probar que T(2)=4 se cumple

$T(2) \leq 3*2\lg 2$?

$4 \leq 6$?

Si, se termina la demostración

Recurrencias

$T(n)=T(n-1)+T(n-2)+1$, $T(1)=O(1)$, $T(2) = O(1)$

Suponer que la solución es de la forma $T(n)=O(2^n)$

Probar que $T(n)\leq c2^n$.

Se supone que se cumple para $n-1$ y se $n-2$ prueba para n

Hipotesis inductiva: $T(n-1) \leq c2^{(n-1)}$ y $T(n-2) \leq c2^{(n-2)}$

Recurrencias

$T(n)=T(n-1)+T(n-2)+1$, $T(1)=O(1)$, $T(2) = O(1)$

Ahora se debe probar que: $T(n) \leq c2^n$

$$T(1) \leq c2^1 \rightarrow 1 \leq 2*c$$

$$T(2) \leq c2^2 \rightarrow 1 \leq 4*c$$

$$T(3) \leq c2^3 \rightarrow 2 \leq 8*c$$

$$T(4) \leq c2^4 \rightarrow 3 \leq 16*c$$

$$T(5) \leq c2^5 \rightarrow 5 \leq 32*c$$

$$T(6) \leq c2^6 \rightarrow 8 \leq 64*c$$

$$T(7) \leq c2^7 \rightarrow 13 \leq 128*c$$

$$T(8) \leq c2^8 \rightarrow 21 \leq 256*c$$

Con $c = 1$, se cumple.

Referencias

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2009. Introduction to Algorithms, Third Edition (3rd ed.). The MIT Press. Chapter 4

Gracias

Próximo tema:

Divide y vencerás