Ordenar recursivamente.

Hay otra forma de ordenar una lista de números en tiempo $O(n \log n)$. En vez de usar una buena estructura de datos, usamos una buena heurística: usar algoritmos recursivos.

Idea. $\underbrace{5, 1, 8, -1, 3,}\ \underbrace{10, 9, 7, 2, 5,}$

Partimos en 2 listas:

$$\underbrace{5, 1, 8, -1, 3,} \qquad \underbrace{10, 9, 7, 2, 5,}$$

Ordenamos cada una

$$-1, 1, 3, 5, 8 \qquad 2, 5, 7, 9, 10.$$

Combinamos las listas ordenadas en una única lista:

$$-1, 1, 2, 3, 5, 5, 7, 8, 9, 10.$$

$\boxed{\cancel{-1}}, \boxed{\cancel{1}}, \boxed{\cancel{3}}, \boxed{\cancel{5}}, \boxed{\cancel{8}}$

$\boxed{\cancel{2}}, \boxed{\cancel{5}}, \boxed{\cancel{7}}, \boxed{\cancel{9}}, \boxed{\cancel{10}}$

$-1, 1, 2, 3, 5, 5, 7, 8, 9, 10$

¿Cuánto tiempo tarda un algoritmo de este estilo?

Sea F la función que cuenta los pasos que nos tardamos.

Entonces se cumple

$$F(n) = D(n) + 2F\left(\frac{n}{2}\right) + J(n).$$

dividir el problema en prob. pequeños.

término anterior

juntar las soluciones chiquitas para obtener la solución grande.

es $O(n)$

$O(n)$.

En resumen para ordenar números recursivamente se cumple.

$$F(n) = 2 \cdot F\left(\frac{n}{2}\right) + O(n)$$

Esto podemos escribirlo como

$$F(n) \leq 2F(n/2) + cn. \qquad 4F\left(\frac{n}{4}\right) + cn + cn$$

$$F(n) \leq 2F(n/2) + cn \leq 2\left(2F\left(\frac{n}{4}\right) + c\frac{n}{2}\right) + cn.$$

$$\leq 2\left(2\left(2\left(F\left(\frac{n}{8}\right) + c\frac{n}{4}\right)\right) + c\frac{n}{2}\right)\right) + cn. = 8F\left(\frac{n}{8}\right) + cn + cn + cn$$

$$= \underbrace{8F\left(\frac{n}{8}\right)} + 3cn.$$

~ tras $\log_2 n$ pasos se llega a una constante

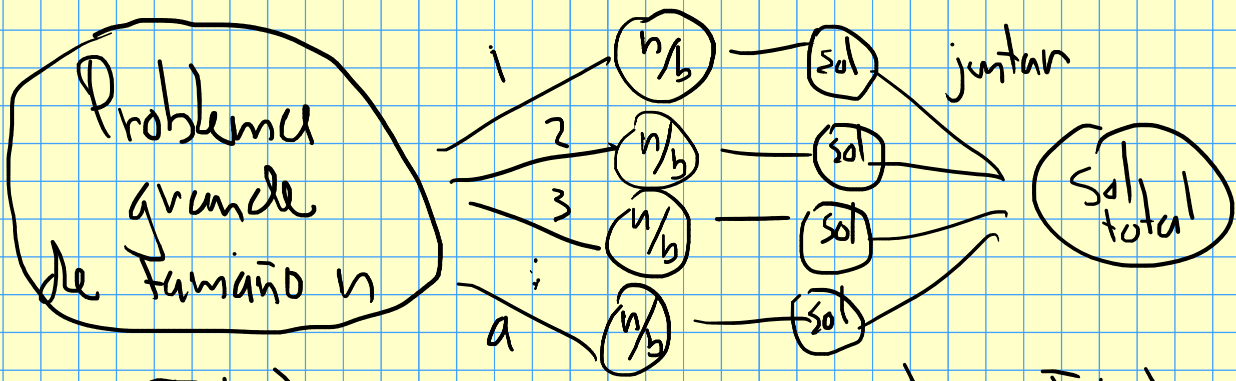
$$\leq \ldots$$

$$\leq 2(2(2(\ldots(2(k + c(2k)) + c(4k) + \cdots) + c\frac{n}{2}) + cn.$$

$$2^{\log_2(n)} \cdot k + \log_2 n \cdot cn = k \cdot n + c \cdot n \cdot \log n.$$

$$= O(n \log n).$$

Método maestro



$$T(n) = D(n) + aT\left(\frac{n}{b}\right) + J(n) = aT\left(\frac{n}{b}\right) + f(n)$$

Teorema. Supongamos que tenemos una función que satisface la siguiente ecuación recursiva:

$$T(n) = aT(n/b) + f(n). \tag{1}$$

en donde $a \geq 1$ y $b > 1$ son constantes y $f(n)$ es una función positiva. Definamos $d = \log_b a = \log a / \log b$, al cual le llamaremos el **exponente crítico**. Entonces, tenemos los siguientes tres casos:

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

2) Si $f(n) = \Theta(n^d \log^k n)$ para alguna $k \geq 0$, entonces $T(n) = \Theta(n^d \log^{k+1} n)$.

3) Si $f(n) = \Omega(n^{d+\epsilon})$ para alguna constante $\epsilon > 0$ y además $f(n)$ satisface la **condición de regularidad** $af(n/b) < cf(n)$ para alguna constante $c < 1$ y n suficientemente grande, entonces $T(n) = \Theta(f(n))$.

Problema La función $f(n)$ satisface la recursión

$$f(n) = f(n-1) + 3 \qquad (\text{para } n \geq 1).$$

Determina el orden de crecimiento de $f(n)$.

Sol. No podemos usar el método maestro pues $n-1$ no es de tamaño n/b para ninguna $b > 1$.

Notemos

$$f(n) = f(n-1)+3 = f(n-2)+3+3 = f(n-3)+3+3+3$$

$$= \ldots = f(0) + \underbrace{3+3+\ldots+3}_{n \text{ veces}} = f(0) + 3n = \Theta(n).$$

Problema. La función $f(n)$ satisface $f(n) \leq f(n-2) + c$

$$f(n) = f(n-2) + O(1)$$

¿Qué podemos decir asintóticamente?

Se puede decir

$$f(n) \leq \underbrace{K}_{f(0) \text{ ó } f(1)} + \left\lfloor \frac{n}{2} \right\rfloor c = O(n).$$

$$f(n) = O(n).$$

Problema La función $f(n)$ satisface

$$f(n) = 2f\left(\frac{n}{3}\right) + \overbrace{n^2 + 3n - \log(n)}^{\Theta(n^2)\ \rightsquigarrow\ g(n)}.$$

¿Qué podemos decir asintóticamente de $f(n)$?

Tomemos $d = \log_3 2 = 0.6309\ldots$

Estamos en el caso 3 de teo maestro.

3) Si $\overset{g}{f}(n) = \Omega(n^{d+\epsilon})$ para alguna constante $\epsilon > 0$ y además $\overset{g}{f}(n)$ satisface la **condición de regularidad** $a\overset{g}{f}(n/b) < c\overset{g}{f}(n)$ para alguna constante $c < 1$ y n suficientemente grande, entonces $\overset{f(n)}{T(n)} = \Theta(\overset{g}{f}(n))$.

$\epsilon = 0.1$

$$f(n) = \Omega\left(n^{0.7309}\right)$$

para alguna c.

Pregunta ¿Se cumple regularidad?

$$a\,g(n/b) < c\,g(n)$$

$$¿\ 2\,g(n/3) < c\,g(n)\,?$$

$$2\left(\frac{n}{3}\right)^2 < c\,n^2$$

$$\frac{2}{9}n^2 < c\,n^2$$

proponemos $c = \frac{1}{2}$

Concluimos por teo. maestro que $f(n) = \Theta(n^2)$.

<u>Problema</u> Estudia asintóticamente la función $T(n)$ que cumple la recursión

$$T(n) = 2T(n/2) + 100n.$$

<u>Sol.</u> Usemos teo. maestros.

1) $a=2$, $b=2$, ambos son >1 ✓

2) $100n$ es positiva para $n \geq 0$ ✓

3) El exp. crítico es $\log_2 2 = 1$.

4) ¿Cómo es el exp. crítico con respecto a el exponente del comportamiento asintótico de $100n$?

$$100n = \Theta(n) = \Theta(n^1).$$

Así, estamos en el caso 2 del teo. maestro

2) Si $f(n) = \Theta(n^d \log^k n)$ para alguna $k \geq 0$, entonces $T(n) = \Theta(n^d \log^{k+1} n)$.

De hecho en nuestro problema se cumple con $k=0$.

$f(n) = 100n = \Theta(n) = \Theta(n \log^0 n)$

Así, por teo. maestro $T(n) = \Theta(n \log n)$.

Problema La función $T(n)$ satisface

$$T(n) = 2f\left(\frac{n}{3}\right) + \sqrt{n} + \sqrt[4]{n} + \sqrt[6]{n} \log n.$$

Sol. Usamos teo. maestro.

1) $2, 3 > 1$ ✓ 2) $\sqrt{n} + \sqrt[4]{n} + \sqrt[6]{n} \log n$. es positiva ✓

3) Tomemos $d = \log_3 2 = 0.6309\ldots$

4) Notamos $\sqrt{n} + \sqrt[4]{n} + \sqrt[6]{n} \log n = \Theta(n^{1/2})$

Así, el exp. crítico es mayor que el de $f(n)$

Estamos en el caso 1. del teo. maestro

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

Se concluye que $T(n) = \Theta(n^{0.6304\ldots})$.