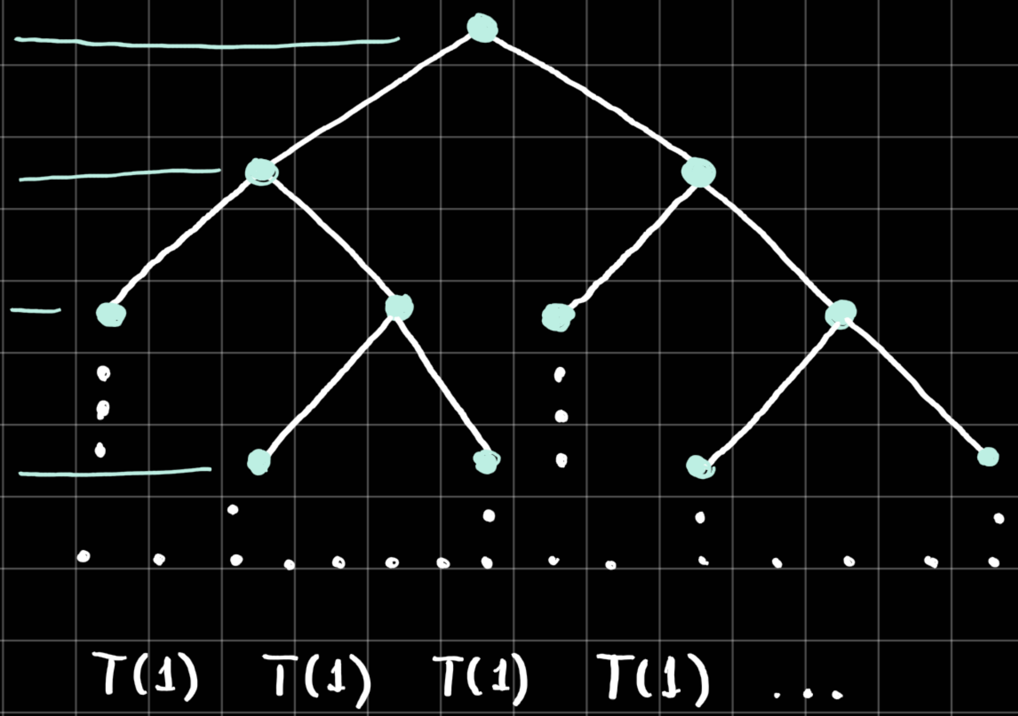
por Isadora de Oliveira

* Anotações Gerais

* Método da Árvore de Recursão

Análise da recorrência $T(n) = 2T\left(\frac{n}{2}\right) + cn$ que é o tempo utilizado no merge-sort.
(vai ser uma árvore binária)

SP = Subproblema



nível	nº SP	tam SP
$\varnothing$	$1 = 2^0$	n
1	$2 = 2^1$	$\frac{n}{2}$
2	$4 = 2^2$	$\frac{n}{4}$
3	$8 = 2^3$	$\frac{n}{8}$
n	2^j	$\frac{n}{2^j}$

Então $n^{\circ} SP(j) = 2^j$ e o $\text{tam } SP(j) = \frac{n}{2^j}$

* Qual o último nível h da árvore?
(note que as chamadas recursivas param quando o tamanho do subproblema é 1)

R: $\text{tam } SP(h) = \frac{n}{2^h} = 1 \rightarrow 2^h = n \rightarrow h = \lg n$

$$\boxed{b^a = c \qquad c = \lg_b^a}$$

* Como comecamos a contar do nível $\emptyset$, o número de níveis da árvore é:

R: $1 + \lg n$

O número de vezes que pode dividir um número por 2 antes de ficar pequeno igual ou menor que 1 é exatamente $\lg n$.

* No nível j o número de subproblema é:

R: 2^j

* O tamanho do vetor de subproblemas é:

R: $\frac{n}{2^j}$

* Qual o trabalho local (não recursivo)?

R: O trabalho não recursivo em um vetor de tamanho m é $c \cdot m$

* Qual o trabalho realizado no nível j?

R: $trab(j) = n^{o} SP(j) \cdot c \cdot tam\,SP(j)$

$$= \cancel{2^j} \cdot c \cdot \frac{n}{\cancel{2^j}} = \boxed{c \cdot n}$$

trabalho uniforme ao longo dos níveis.

* Qual o trabalho total dado pela soma do trabalho por nível da árvore.
(ao longo do número de níveis da árvore)

$$R: T(n) = \sum_{j=0}^{\lg n} trab(j) = \sum_{j=0}^{\lg n} cn = cn(1+\lg n)$$

nível

$$= cn\lg n + cn$$
$$= O(n\lg n)$$

Análise Genérica

Para a recorrência $T(n) = a\left(\frac{n}{b}\right) + c\cdot n^d$

* Número Subproblema no nível j da árvore: a^j

* Tamanho de um subproblema no nível j: $\frac{n}{b^j}$

* $trab\,SP(j) \leq c \cdot tam\,SP(j)^d = c\left(\frac{n}{b^j}\right)^d$

* Trabalho no nível j:

$$trab(j) = num\,SP(j) \cdot trab\,SP(j)$$
$$= a^j \cdot c \cdot \left(\frac{n}{b^j}\right)^d = \frac{c \cdot a^j}{b^{dj}} n^d = c \cdot n^d \left(\frac{a}{b^d}\right)^j$$

* Quantos níveis tem a arvore de recursão?

$$tam\,SP(h) = \frac{n}{b^h} = 1 \rightarrow b^h = n \rightarrow h = \log_b n$$

* Qual o trabalho total realizado?

$$T(n) = \sum_{j=0}^{h} trab(j) = \sum_{j=0}^{h} cn^d \left(\frac{a}{b^d}\right)^j = cn^d \sum_{j=0}^{\lg_b n} \left(\frac{a}{b^d}\right)^j$$

Chegamos a soma dos termos de uma PG com razão

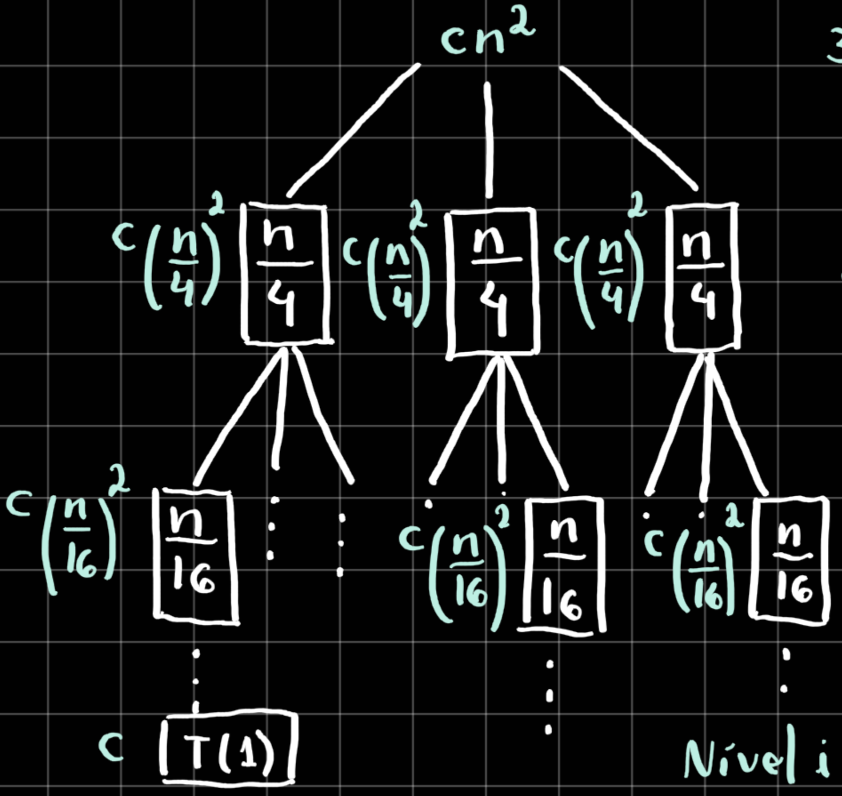
$$\left(\frac{a}{b^d}\right) \quad \text{se} < 1 \quad S = \frac{1}{1-r} = \frac{1}{1-a/b} = \frac{1}{a/b} = \frac{b}{a}$$

$$\text{se} > 1 \quad S = \frac{r^{n+1} - 1}{r - 1}$$

* Resumo Árvore de Recursão

$T(n) = 3T\left(\frac{n}{4}\right) + \Theta(n^2)$ - Quero descobrir $T(n) = O(\ldots)$

Consideramos n potência de 4 para termos apenas nº inteiro



(nó) nº Sub. Prob.	tam. SP	custo nível
3^0	$\frac{n}{4^0}$	$3^0 \cdot c\left(\frac{n}{4^0}\right)^2 = cn^2$
3^1	$\frac{n}{4^1}$	$3c\left(\frac{n}{4}\right)^2$
3^2	$\frac{n}{4^2}$	$9c\left(\frac{n}{16}\right)^2$
Nível i: 3^i	$\frac{n}{4^i}$	$3^i c\left(\frac{n}{4^i}\right)^2$

nº $SP = a^i$

tam $SP = \frac{n}{b^i}$

custo nível $= a^i \cdot c\, \frac{n}{b^i}$ → aplica notação

altura $= \frac{n}{b^H} = 1$

(O 1 porque vai executar a arvore recursão até 1)

Altura $= \frac{n}{4^H} = 1 \to 4^H \cdot n$ (aplica lg)

$H = \lg_4 n$

$n \lg_b a \to b^n = a$

custo total $= \sum_{i=0}^{\text{altura}}$ custo nível

$\frac{a}{b} < 1 \to S = \frac{1}{1-r}$

$\frac{1}{a/b} = \frac{b}{a}$

$\frac{a}{b} > 1 \to S = \frac{r^{n\text{o}+1} - 1}{r-1}$

Nº folhas no ultimo nível $T(1) = a^{\text{altura}}$

$3^{\lg_4 n} = n^{\lg_4 3}$

$\Theta(n^{\lg_4 3})$

Teorema Mestre

$$\sum_{i=0}^{\lg_4 n} 3^i c\left(\frac{n}{4^i}\right)^2 \Rightarrow c \cdot n^2 \sum_{i=0}^{\lg_4 n} \left[\frac{3}{16}\right]^i \rightarrow \frac{3^i}{16^i}$$

$$= cn^2 \cdot \frac{16}{13} \in O(n^2)$$

$$S = \frac{1}{1-r}$$

$$S = \frac{1}{1-3/16}$$

$$S = \frac{1}{13/16} = \frac{16}{13}$$