Teorema Master

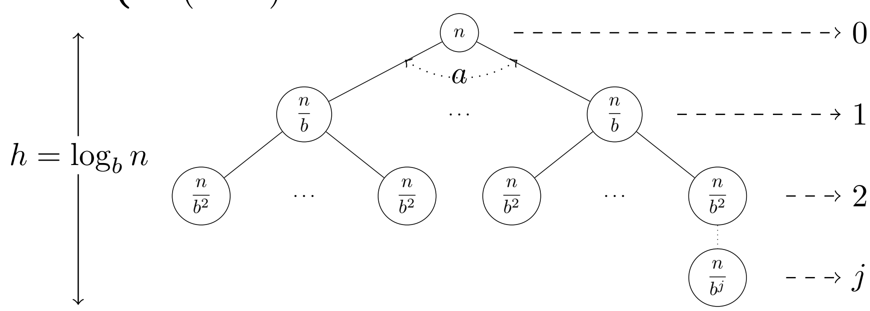
$T(n) = a \cdot T\left(\frac{n}{b}\right) + \mathcal{O}\left(n^d\right)$

a = numărul de subprobleme, $a > 0$

b = rata de descreştere a dimensiunii subproblemelor, $b > 1$

d = exponentul din $Div(n) + Comb(b) \in \mathcal{O}\left(n^d\right)$, $d > 0$

$T(n) = \begin{cases} \mathcal{O}\left(n^d \cdot \log(n)\right) & a = b^d \\ \mathcal{O}\left(n^d\right) & a < b^d \\ \mathcal{O}\left(n^{\log_b a}\right) & a > b^d \end{cases}$



a^j = numărul de subprobleme la nivelul j

$\frac{n}{b^j}$ = dimensiunea subproblemelor de la nivelul j

Avem $a^{\log_b n} = n^{\log_b a}$ frunze.

$$\begin{aligned} T(\text{nivel}j) &\leq a^j \cdot \mathcal{O}\left(\left(\frac{n}{b^j}\right)^d\right) \\ &= a^j \cdot c \cdot \left(\frac{n}{b^j}\right)^d \\ &= c \cdot n^d \cdot \left(\frac{a}{b^d}\right)^j \end{aligned}$$

Avem:

$$\begin{cases} (i) & T(1) \le c \\ (ii) & T(n) \le a \cdot T\left(\dfrac{n}{b}\right) + c \cdot n^d \end{cases}$$

$$\begin{aligned} \text{Total} &= \sum_{j=0}^{\log_b n} T\,(\text{nivel } j) \\ &\le \sum_{j=0}^{\log_b n} c \cdot n^d \cdot \left(\frac{a}{b^d}\right)^j \\ &= c \cdot n^d \cdot \underbrace{\sum_{j=0}^{\log_b n} \left(\frac{a}{b^d}\right)^j}_{\underset{\text{serie geom.}}{\boxed{\sum_{j=0}^{\log_b n} r^j}} \quad \underset{\text{raţia}}{\boxed{r = \frac{a}{b^d}}}} \end{aligned}$$

$T(n) \leq c \cdot n^d \cdot \left(\sum_{j=0}^{\log_b n} r^j \right), \ r = \dfrac{a}{b^d}$
Sume de serii geometrice
$1 + r + r^2 + \cdots + r^k = \begin{cases} \dfrac{r^{k+1} - 1}{r - 1}, & \text{dacă } r \neq 1 \\ k + 1, & \text{dacă } r = 1 \end{cases}$

(1) dacă $r = 1$, $\sum_{j=0}^{\log_b n} r^j$ = numărul de termeni $= k + 1$

(2) dacă $r < 1$,
$sum_{j=0}^{\log_b n} r^j \leq \dfrac{1}{1 - r}$

(3) dacă $r > 1$,
$sum_{j=0}^{\log_b n} r^j \leq r^k \left(1 + \dfrac{1}{r - 1} \right)$

Obs.: Atât $\dfrac{1}{1 - r}$, ąt şi $1 + \dfrac{1}{r - 1}$, sunt independente de k (cazurile (2) şi (3)).
Cazul (1) $(r = 1)$ $\underline{\underline{a = b^d}}$:

$$\begin{aligned} T(n) &\leq c \cdot n^d \cdot (\log_b n + 1) \\ &= \mathcal{O}\left(n^d \cdot \log n\right) \end{aligned}$$

Cazul (2) $(r < 1)$ $\underline{\underline{a < b^d}}$:

$$\begin{aligned} T(n) &\leq c \cdot n^d \cdot \underbrace{\left(\dfrac{1}{1 - \dfrac{a}{b^d}} \right)}_{\text{constant}} \\ &= \mathcal{O}\left(n^d\right) \end{aligned}$$

Cazul (3) $(r > 1)$ $\underline{\underline{a > b^d}}$:
$T(n) \leq c \cdot n^d \cdot \underbrace{\left(1 + \dfrac{1}{r - 1} \right)}_{\text{constant}} \cdot r^{\log_b n}$

Din $r^{\log_b n} = \left(\dfrac{a}{b^d} \right)^{\log_b n} = \dfrac{a^{\log_b n}}{\left(b^{\log_b n}\right)^d} = \dfrac{a^{\log_b n}}{n^d}$ rezultă:
$T(n) \leq \mathcal{O} \underbrace{\left(a^{\log_b n}\right)}_{\underline{\underline{\text{nr. de frunze}}}}$ Amintim că numărul de frunze este: $a^{\log_b n} = n^{\log_b a}$.

Căutare binară

```
CautBin(x, A[1..n])                          // vectorul A este ordonat descrescător
Θ(1)   {  if x = A[n/2] then
             gata
          else
T(n/2) {    if x < A[n/2] then
               CautBin(x, A[1..n/2 − 1])
            else
               CautBin(x, A[n/2 + 1..n])
            endif
          endif
```

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

$\underline{a = 1},\ \underline{b = 2},\ \underline{d = 0}$

$a \cdot T\left(\frac{n}{2}\right) + \Theta(n^d)$

$\underline{a = b^d} \xrightarrow[\text{caz 1}]{} \underline{\underline{T(n) = \mathcal{O}(\log n)}}$

Merge Sort

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

$\underline{a = 2},\ \underline{b = 2},\ \underline{d = 1}$

$\underline{a = b^d} \xrightarrow[\text{caz 1}]{} \underline{\underline{T(n) = \mathcal{O}(n \log n)}}$

Quick Sort

`QuickSort(`$A[1..n]$`)`

1. $k \leftarrow$ `Partitie(`A`)` .... *Divide* $= \Theta(n)$
2. `QuickSort(`$A[1..k-1]$`)` .... *apel recursiv 1*
3. `QuickSort(`$A[k+1..n]$`)` .... *apel recursiv 2*

.... *Conquer = 0*

Cazul cel mai favorabil

$k = \frac{n}{2}$ (la toate!)

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

$\underline{a = 2}, \ \underline{b = 2}, \ \underline{d = 1}$

$\underline{a = b^d} \xrightarrow[\text{caz 1}]{} \underline{\underline{T(n) = \mathcal{O}(n \log n)}}$

Înmulţirea a două numere de lungime n

- Algoritmul ,,direct" $\mathcal{O}\left(n^2\right)$
- Algoritmul ,,Divide et Impera" (Gauss/Karatsuba)

$$x = \boxed{x_L \mid x_R} = 2^{\frac{n}{2}} \cdot x_L + x_R$$

$$y = \boxed{y_L \mid y_R} = 2^{\frac{n}{2}} \cdot y_L + y_R$$

$$x \cdot y = 2^n \cdot \underbrace{x_L \cdot x_R} + 2^{\frac{n}{2}} \left(\underbrace{x_L \cdot y_R} + \underbrace{x_R \cdot y_L} \right) + \underbrace{x_R \cdot y_R}$$

$T(n) = 4 \cdot T\left(\frac{n}{2}\right) + \mathcal{O}(n)$

$\underline{a = 4},\ \underline{b = 2},\ \underline{d = 1}$

$\underline{a > b^d} \xrightarrow[\text{caz 3}]{} T(n) = \mathcal{O}(n^{\log_2 4}) = \underline{\underline{\mathcal{O}\left(n^2\right)}}$

- De la 4 apeluri recursive la 3

$T(n) = 3 \cdot T\left(\frac{n}{2}\right) + \mathcal{O}(n)$

$\underline{a = 3},\ \underline{b = 2},\ \underline{d = 1}$

$\underline{a > b^d} \xrightarrow[\text{caz 3}]{} T(n) = \mathcal{O}(n^{\log_2 3}) = \underline{\underline{\mathcal{O}\left(n^{1,59}\right)}}$