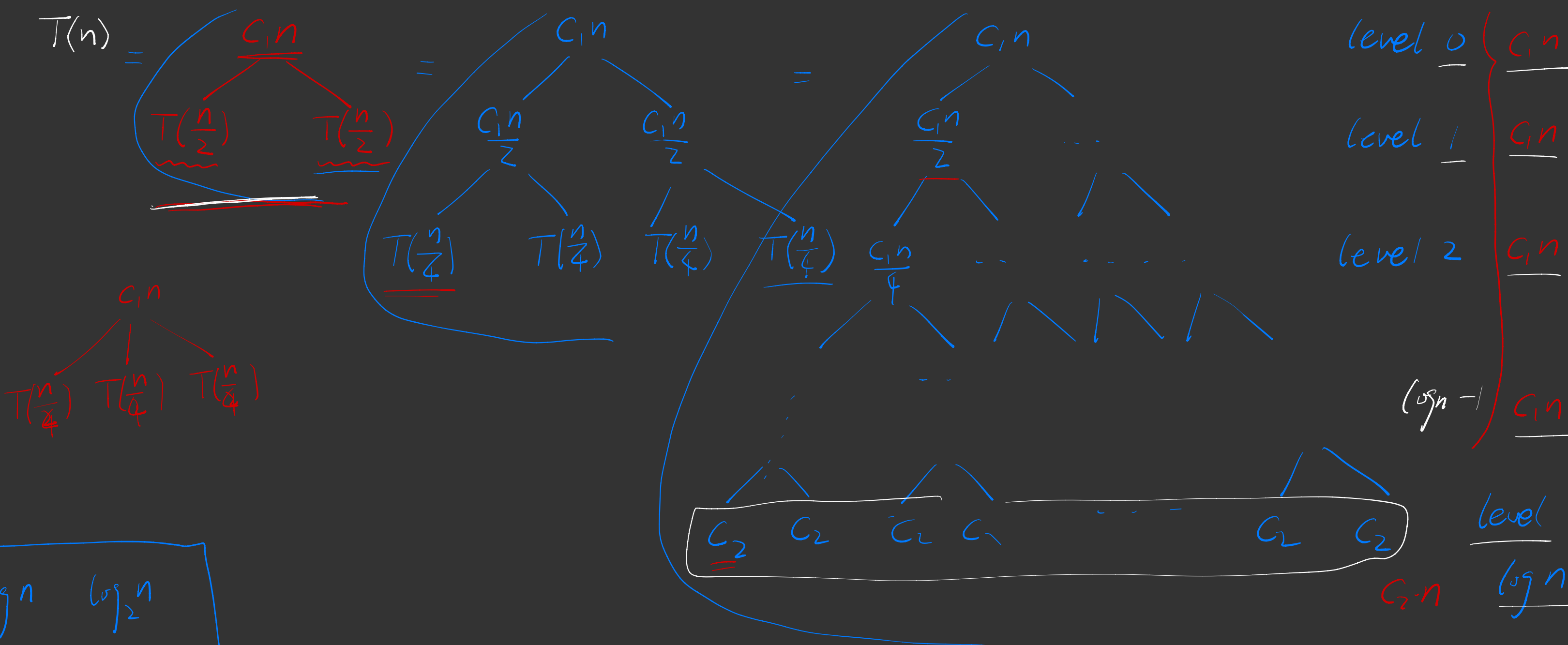
Merge Sort: $\begin{cases} T(n) = \overset{3}{\boxed{2}}\, T(n/\overset{n/4}{\boxed{2}}) + \boxed{c_1 n} \\ T(1) = c_2 \end{cases}$ (*)

$T(\frac{n}{2}) = 2\,T(\frac{n}{4}) + c_1 \cdot \frac{n}{2}$

✓ Recurrence tree

✓ Induction

Master theorem




$\log n \quad \log_2 n$

$T(n)$ = Contribution from all levels = $c_1 n \cdot \log n + c_2 n = \Theta(n \log n)$

Good guess about $T(n)$

Prove it formally by induction

$\boxed{T(n) = O(n\log n)}$

$\boxed{T(n) \geq an\log n}$

$$\begin{cases} T(n) = 2T(\frac{n}{2}) + c_1 n \\ T(1) = c_2 \end{cases}$$

$T(n) \leq a \cdot n\log n + b$

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

(*) positive

Statement: $T(n) \leq a \cdot n\log n + b$ for some positive constants $a, b$

Base case: $T(1) \leq a \cdot n\log n + b$ when $n = 1$

$c_2 = 0 + b$

it holds as long as $\boxed{b \geq c_2}$

$b = c_2$

$a = c_1 + c_2$

Induction Step: Assuming (*) holds for everything below $n$
prove (*) at $n$.

$$T(n) = 2T\left(\frac{n}{2}\right) + c_1 n \overset{(IH)}{\leq} an(\log n - 1) + 2b + c_1 n \overset{?}{\leq} an\log n + b$$

$b + c_1 n \overset{?}{\leq} an$

$$T\left(\frac{n}{2}\right) \leq a \cdot \frac{n}{2}\log\left(\frac{n}{2}\right) + b = \frac{an}{2}(\log n - 1) + b \quad (IH)$$

it holds as long as $\boxed{a \geq b + c_1}$

Binary Search: Input $A = \langle a_1 \ldots a_n \rangle$ in nondecreasing order and an integer $x$

Output $i$ with $a_i = x$ if $x$ is in $A$; nil if $x$ is not in $A$

If $n=1$, just compare $a_1$ with $x$

[ Compare $a_{\frac{n}{2}}$ with $x$

case 1 same: output $\frac{n}{2}$

2 $a_{\frac{n}{2}} > x$: recursively Binary Search the first half

3 $a_{\frac{n}{2}} < x$: - - - - - - - - - - second half

$$T\left(\frac{n}{2}\right) = T\left(\frac{n}{4}\right) + c_1$$

$$\begin{cases} T(n) = 1 \cdot T\left(\frac{n}{2}\right) + \overset{c_1}{\Theta(1)} & (*) \\ T(1) = \Theta(1)^{c_2} \end{cases}$$

$$\begin{aligned} T(n) &= T\left(\frac{n}{2}\right) + \overset{\Theta(1)}{c_1} \\ &= T\left(\frac{n}{4}\right) + \overset{\Theta(1)+\Theta(1)}{2c_1} \\ &= T\left(\frac{n}{8}\right) + 3c_1 \\ &\;\vdots \\ &= T(1) + c_1 \log n = c_2 + c_1 \log n \end{aligned}$$

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

# Matrix Multiplication:

Input: two $n \times n$ matrices $A = (a_{ij})$ $B = (b_{ij})$

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ & \ddots & & \vdots \\ a_{n1} & \cdots & & a_{nn} \end{pmatrix}$$

Compute $\underline{C = AB}$ $C = (c_{ij})$ $n \times n$

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

$$T(n) = \boxed{8} T\left(\frac{n}{2}\right) + \Theta(n^2)$$

$$T(1) = \Theta(1)$$

$\log_2 8 = 3$ $\Theta(n^{3})$

Brute force: $\Theta(n^3)$

Divide and conquer: Smaller tasks: multiplying smaller matrices.

$$A = \begin{pmatrix} A_{1,1} & A_{1,2} \\ A_{21} & A_{22} \end{pmatrix} \qquad B = \begin{pmatrix} B_{1,1} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

$A_{1,1} \cdots B_{22}$,

$\left(\frac{n}{2}\right) \times \left(\frac{n}{2}\right)$ matrices.

$$C = \begin{pmatrix} C_{1,1} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$$

$C_{11} = \boxed{A_{11} \cdot B_{11}} + \boxed{A_{12} \cdot B_{21}}$ $\left(\frac{n}{2} \cdot \frac{n}{2}\right) \cdot 4$

8 recursive calls

$C_{12} = A_{11} B_{12} + A_{12} B_{22}$

$C_{21}$ $C_{22}$

Strassen's algorithm:

$P_1 = A_{11}(B_{12} - B_{22})$

$P_2 = (A_{11} + A_{12}) B_{22}$

$P_3 = (A_{21} + A_{22}) B_{11}$

$P_4 = A_{22}(B_{21} - B_{11})$

$P_5 = (A_{11} + A_{22})(B_{11} + B_{22})$

$P_6 = (A_{12} - A_{22})(B_{21} + B_{22})$

$P_7 = (A_{11} - A_{21})(B_{11} + B_{12})$

$\frac{n}{2} \times \frac{n}{2}$

$C_{11} = P_5 + P_4 - P_2 + P_6$

$C_{12} = P_1 + P_2$

$C_{21} = P_3 + P_4$

$C_{22} = P_5 + P_1 - P_3 - P_7$

$P_1 + P_2 = A_{11}B_{12} + A_{12}B_{22} = C_{12}$

$8\,T(\frac{n}{2})$

$$\begin{cases} T(n) = \cancel{7} \cdot T(\frac{n}{2}) + \Theta(n^2) \\ T(1) = \Theta(1) \end{cases}$$

$\Theta(n^{3 = \log_2 8})$

$\Theta(n^{\log_2 7}) = \Theta(n^{2.81})$

MM constant $\omega$

$n^\omega$ is the time needed for MM.

$\boxed{n^2} \leftrightarrow n^{2.3728596} \cdots n^{2.81}\ n^3$

Josh Alman