Aprob

if $n=1$, $T(1) \le O(1)$

if $n>1$, $T(n) \le O(1) + O(n) + T(\lfloor \frac{n}{2} \rfloor) + T(\underbrace{n - \lfloor \frac{n}{2} \rfloor}_{\lceil \frac{n}{2} \rceil})$ $+O(n)$

$$T(n) \le \begin{cases} O(1) & \text{if } n=1 \\ \cancel{O(1)+}\, O(n) + T(\lfloor \frac{n}{2} \rfloor) + T(\lceil \frac{n}{2} \rceil), & o/w \end{cases}$$

By defn of Big-oh, $\exists$ constants c_1 & c_2,

$$T(n) \le \begin{cases} c_1 & \text{if } n=1 \\ c_2 n + T(\lfloor \frac{n}{2} \rfloor) + T(\lceil \frac{n}{2} \rceil), & o/w \end{cases}$$

Use a $c = \max(c_1, c_2)$,

$$T(n) \le \begin{cases} c & \text{if } n=1 \\ cn + T(\lfloor \frac{n}{2} \rfloor) + T(\lceil \frac{n}{2} \rceil), & o/w \end{cases}$$

for asymptotic bound of $T(n)$, it enough

$\Rightarrow T(\lfloor x \rfloor) \to T(x)$ & $T(\lceil x \rceil) \to T(x)$

Recurrence Relation

$$T(n) \le \begin{cases} c, & \text{if } n=1 \\ cn + 2T(\frac{n}{2}), & o/w \end{cases}$$

Strategies to solve Recurrence relation:

① "Unroll" the recursion; identify a pattern; Use the pattern to reach a solution.

Recurrence tree method

② ← substitution method
"Guess ~~a~~ the answer & verify by induction on n

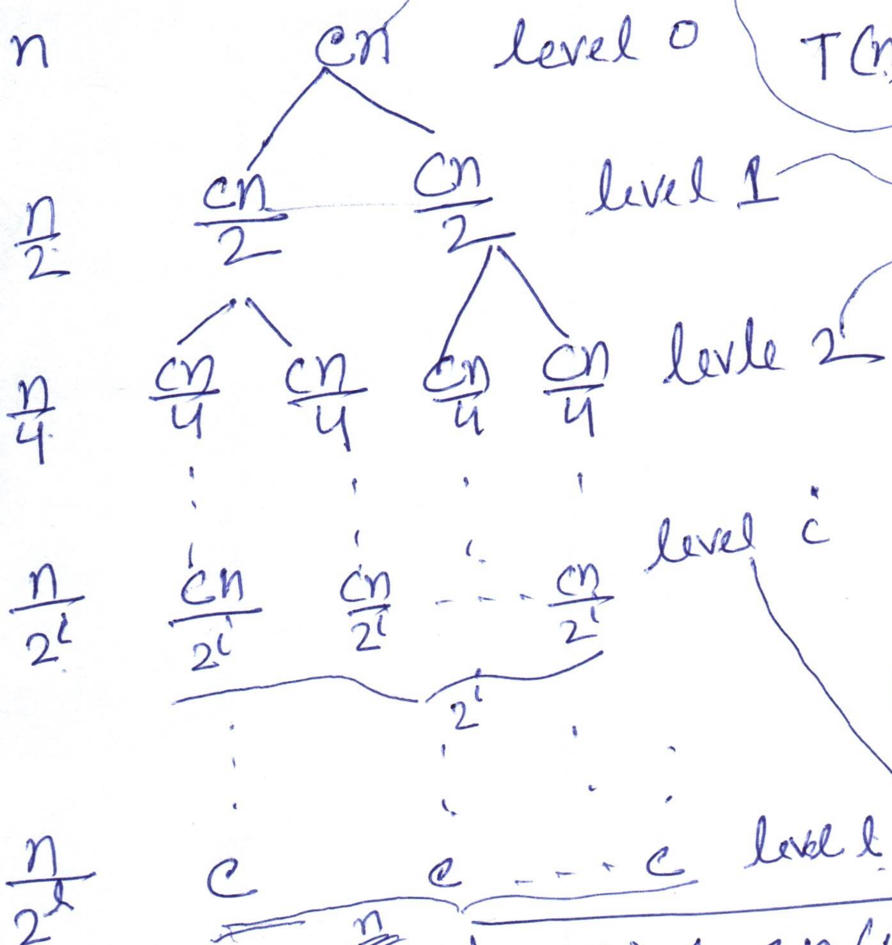
~~St~~ $T(n) \le \begin{cases} c, & \text{if } n = 1 \\ cn + 2T(\frac{n}{2}), & o/w \end{cases}$

<u>lemma:</u> $T(n) \le cn\log_2 n + cn$ ← $(\le O(n\log n))$

$\Rightarrow$ Merge Sort runs in $O(n\log n)$ time.

<u>Strategy 1:</u> "unroll" + identify a "patten" + use the "pattern"

$\boxed{\text{Assume } n \text{ is a power of } 2}$



$T(n) \le cn + 2T(\frac{n}{2})$

$\le cn + 2\cdot(c\cdot\frac{n}{2} + 2T(\frac{n}{4}))$

$= cn + cn + 4T(\frac{n}{4})$

$\le cn + cn + 4\cdot(c\cdot\frac{n}{4} + 2T(\frac{n}{8}))$

$= cn + cn + cn + 8T(\frac{n}{8})$

contribution from level 0: cn

" " " 1:

$\frac{cn}{2} + \frac{cn}{2} = cn$

" " " 2:

$\frac{cn}{4} \times 4 = cn$

$\frac{cn}{2^i} \times 2^i = cn$

$\frac{n}{2^l} = 1$

$\Rightarrow n = 2^l$

$\Rightarrow l = \log_2 n$

$T(n) \le cn(\#\text{levels})$

$= cn(l+1)$

$= cn(\log_2 n + 1)$

$= cn\log_2 n + cn$ □

Strategy 2: Given the answer + verify by induction on n.

Given $T(n) \le cn\log_2^n + cn$. $\rightarrow (*)$ $\quad [T(1) \le c$, if $n=1]$

Base Case: $n=1$, $T(1) \le c \cdot 1 \cdot \log_2^1 + c \cdot 1$

$T(1) \le c$

I.H. Assume $n \Rightarrow \frac{n}{2}$, $T(n) \le cn\log_2^n + cn$ holds.

$\Rightarrow \forall n,\ 1 \le n \le \frac{n}{2} \longrightarrow (*)$ holds.

$T(n) \le cn\log_2^n + cn$

$$T(n/2) \le c \cdot \frac{n}{2} \cdot \log_2^{\frac{n}{2}} + c \cdot \frac{n}{2}$$

$$= \frac{cn}{2}\left(\log_2^{\frac{n}{2}} + 1\right)$$

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

$$= \frac{cn}{2}\log_2^n$$

I.S. $T(n) \le cn + 2T\left(\frac{n}{2}\right)$

By I.H. $\longrightarrow$ $\le cn + 2 \cdot \left(\frac{cn}{2}\log_2^n\right)$

$= cn + cn\log_2^n \quad \square$