Nov 3, 2014

MergeSort (a, n)

$O(1)$ $\begin{cases} \text{If } n=1 & \text{return } a_1; \\ \text{If } n=2 & \text{return } \min(a_1, a_2); \max(a_1, a_2); \end{cases}$

$O(n)$ $\begin{cases} a_L = a_1, \ldots, a_{\lfloor \frac{n}{2} \rfloor} \\ a_R = a_{\lfloor \frac{n}{2} \rfloor + 1}, \ldots, a_n \end{cases}$

$T(\lfloor \frac{n}{2} \rfloor)$ $\quad T(n - \lfloor \frac{n}{2} \rfloor)$

return MERGE(MergeSort $(a_L, \lfloor \frac{n}{2} \rfloor)$,
MergeSort $(a_R, n - \lfloor \frac{n}{2} \rfloor)$)

$O(n)$

$T(n) \rightarrow$ max run time of MergeSort on any n numbers.

If $n \le 2$, $T(n) \le O(1)$

otherwise

$$T(n) \le O(1) + O(n) + O(n) + T(\lfloor \tfrac{n}{2} \rfloor) + T(n - \lfloor \tfrac{n}{2} \rfloor)$$
$$= O(n) + T(\lfloor \tfrac{n}{2} \rfloor) + T(n - \lfloor \tfrac{n}{2} \rfloor)$$

$$T(n) \le \begin{cases} O(1) & \text{if } n \le 2 \\ O(n) + T(\lfloor \frac{n}{2} \rfloor) + T(n - \lfloor \frac{n}{2} \rfloor) \end{cases}$$

$$T(n) \le \begin{cases} c_1 & \text{if } n \le 2 \\ c_2 n + T(\lfloor \frac{n}{2} \rfloor) + T(n - \lfloor \frac{n}{2} \rfloor) \end{cases}$$

can ignore $\lfloor\ \rfloor$ & $\lceil\ \rceil$ for asymp. analysis

$$T(n) \le c \qquad n \le 2 \qquad c = \max(c_1, c_2$$
$$\le cn + 2T\left(\frac{n}{2}\right) \quad o/w$$

Lemma: $T(n) \le c\, n \log_2 n$

($\Rightarrow$ MergeSort runs in $O(n \log n)$ time)

$\rightarrow$ No better generic sorting algorithms than $O(n \log n)$

$\rightarrow$ Any comparison based algo needs $\Omega(n \log n)$ comparisons.

$\rightarrow$ Can do better if e.g. $a_i \in \{1, \ldots, n\}$

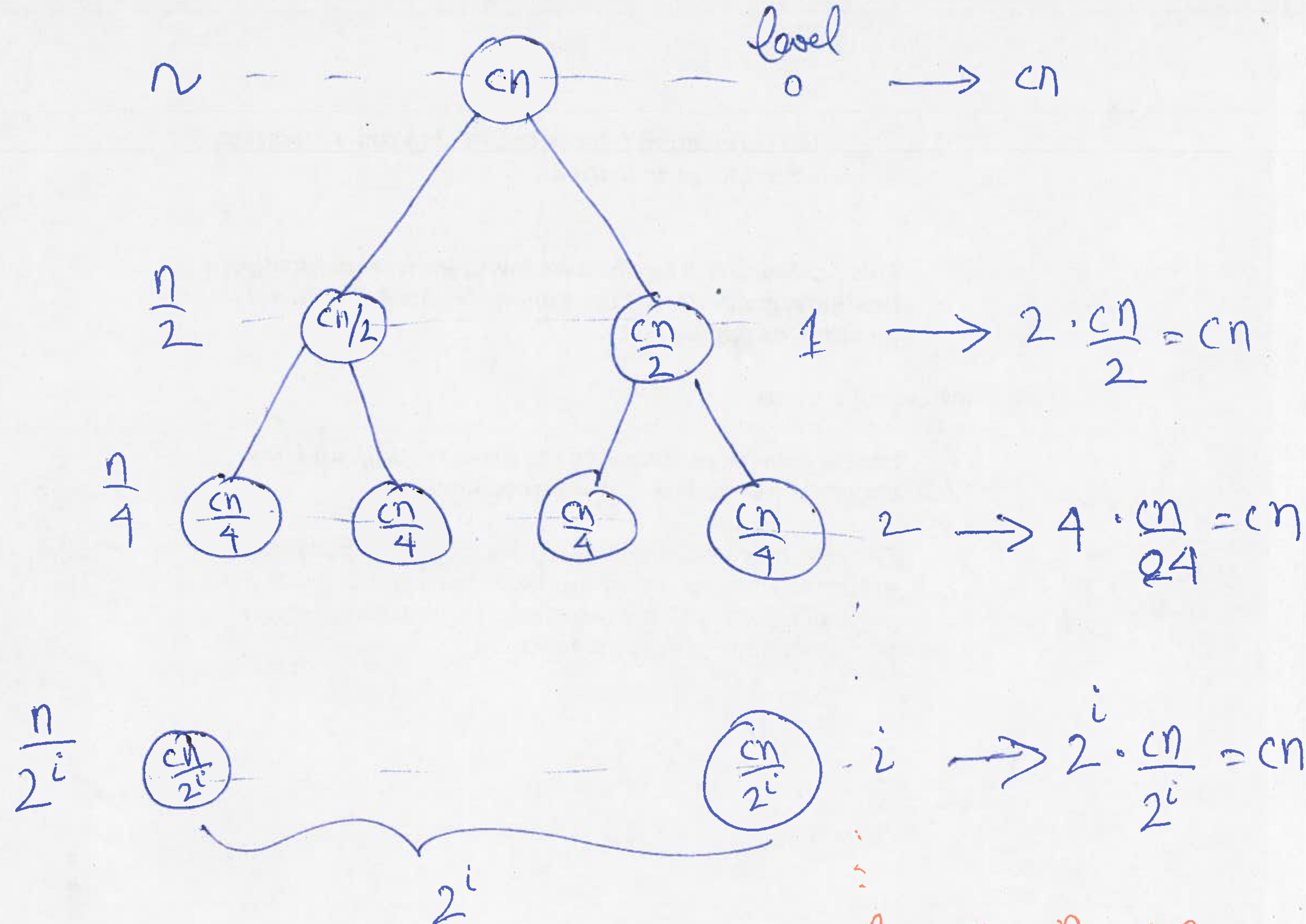
$\Rightarrow O(n)$ sorting (Radix sort)

Strategies to solve recurrences

① Unroll the recursion for 1-2 steps & identify a pattern

② Guess & verify (by induction on n)
($c\, n \log_2 n$)

$$T(n) \le cn + \underbrace{2T\left(\frac{n}{2}\right)}$$
$$\le 2\left(c\frac{n}{2} + 2T\left(\frac{n}{4}\right)\right)$$
$$= cn + cn + 4T\left(\frac{n}{4}\right)$$



$$\Rightarrow T(n) \le cn \underbrace{(\#\text{levels})}_{\le \log_2 n} \le l+1$$

$$\le cn \log_2 n .$$

l s.t. $\dfrac{n}{2^l} \le 2$

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

$$\Rightarrow l = \log_2 \frac{n}{2}$$

$$= \log_2 n - 1$$

$$\boxed{\Rightarrow l+1 \le \log_2 n}$$

(2) Guess & verify

$T(n) \le c\, n \log_2 n$ $\quad$ ($n \to$ power of 2)

Prove by induction:

Base case $n = 2$ $\quad T(2) \le c \overset{?}{\le} c \cdot 2 \log_2 2$

$= 2c$ ✓

Assume true til $\frac{n}{2}$ $\left(T\left(\frac{n}{2}\right) \le c \frac{n}{2} \log_2 \frac{n}{2} \right)$

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

from I.H. $\rightarrow$ $\le cn + 2\left(\frac{cn}{2} \log_2 \frac{n}{2}\right)$

$= \cancel{cn + 2\left(\frac{cn}{2} + \log\right.}$

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

$$= cn + cn \log_2 n - cn$$

$$= cn \log_2 n.$$

✓