NOTE BOOK

Rokshana Ahmed

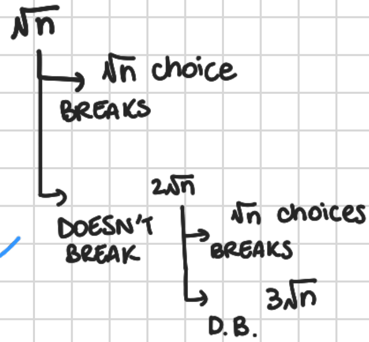
Algorithms

DATE: 12/4/2022

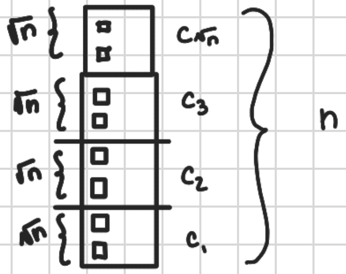
1 PHONE

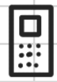
2 PHONE

3 PHONE



$O\left(\frac{n}{t} + t\right)$





1 PHONE → $O(n)$

2 " → $O(\sqrt{n})$

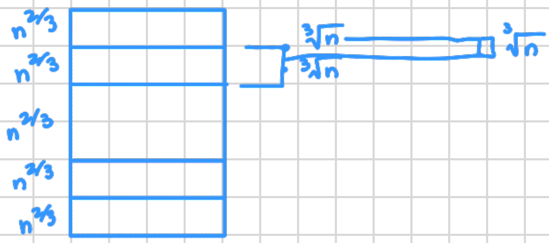
3 " → $O(\sqrt[3]{n^2})$

K " → $O(K \sqrt[K]{n})$

K−1

$$\boxed{\frac{n}{2^{K-1}}}$$

NOTE: we throw from the last floor of the chunk



$K = \log n$ PHONES → $O(K \sqrt[K]{n})$

$$\sqrt[K]{n} = n^{1/K} = \left(2^{\log_2 n}\right)^{1/K} =$$

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

$$O\left(\log_2 n \sqrt[\log_2 n]{n}\right) = O(\log_2 n)$$

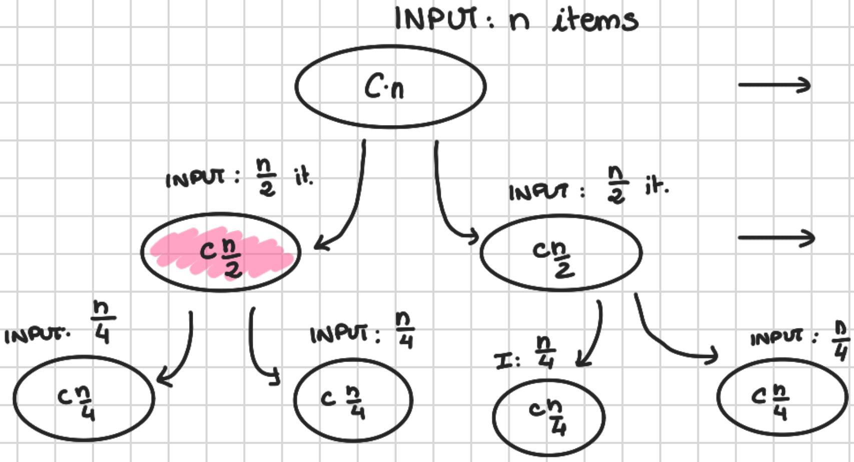
We looked at MERGESORT on $n = 2^t$, for t positive integer, elements.

We proved that the runtime of MERGESORT is $T(n)$, and it satisfies:

$$\exists\, c > 0, \quad T(n) \le \begin{cases} 2T\left(\frac{n}{2}\right) + c_n & \forall n > 2 \\ T(n) \le c & n \in \{0, 1, 2\} \end{cases}$$

APPROACH 1

Unroll the recurrence by looking at its first few steps.



$\longrightarrow \quad L_0 = cn$

$\longrightarrow \quad L_1 = 2\cdot\left(c\,\frac{n}{2}\right) = cn$

$L_2 = 2^2\left(c\,\frac{n}{4}\right) = cn$

In the iTH level, there will be 2^i calls.
The generic call of the ith level is going to take time $\frac{cn}{2^i}$ (plus whatever its own calls will cost).

The total runtime for level "i" is going to be

$$2^i \cdot \left(\frac{cn}{2^i}\right) = cn = O(n)$$

Since there are $\log_2 n$ levels, the total runtime is going to be $O(n \log n)$

APPROACH 2

HOW TO BOUND $T(n)$?

1. Guess a particular bound ($T(n) \leq n^2$, $T(n) \leq n$).

Suppose we believe (and would like to check) that $T(n) \leq a \cdot n \log_2 n$, for some constant $a > 0$.

R2

$$\exists\, c > 0, \quad T(n) \leq \begin{cases} 2T\left(\frac{n}{2}\right) + c_n & \forall n > 2 \\ T(n) \leq c & n \in \{0, 1, 2\} \end{cases}$$

The case $T(2)$ clearly holds, if $2a \geq c$,

$T(2) \leq c$

We want to claim that $T(2) \leq a\, 2 \log_2 2 = 2e$.
The base case holds

Suppose, now, that $n \geq 3$. By induction, we know that $T(m) \leq 2T\left(\frac{m}{2}\right) + c\,m \quad \forall m \leq n-1$.

We want to prove the inequality for n.

$$T(n) \overset{R_2}{\leq} 2T\left(\frac{n}{2}\right) + c_n$$

I.H. (Induction Hyphotesis) $\rightarrow$

$$\leq \cancel{2}\left(a\,\frac{n}{\cancel{2}}\log_2\frac{n}{2}\right) + c_n$$

$$= a\,n\left(\log_2(n) - 1\right) + c_n$$
$$= a\,n\log_2 n - a\,n + c\,n$$
$$= a\,n\log_2 n + (c-a)\,n$$

$a \geq c \rightarrow \quad \leq a\,n\log_2 n \quad \checkmark$

L: Thus, if $a \geq c$, we have that $T(n) \leq a\,n\log_2 n$.

C: $T(n) \leq c\,n\log_2 n$

R_2

$$\exists\, c > 0, \quad T(n) \leq \begin{cases} 2T\left(\frac{n}{2}\right) + c_n & \forall n > 2 \\ c & n \in \{0, 1, 2\} \end{cases}$$

$$\begin{array}{r} 12345\ + \\ 67193\ = \\ \hline 79538 \end{array} \qquad \begin{array}{r} 100110\ + \\ 11011\ = \\ \hline 1000001 \end{array}$$

THE RUNTIME OF THIS ALGORITHM IS $O(n)$ IF YOU SUM UP TWO NUMBERS OF n DIGITS EACH

$$\begin{array}{r} 324\ \times \\ 156\ = \\ \hline 1924 \\ 1620 \\ 324 \\ \hline 50524 \end{array}$$

THE RUNTIME OF THIS ALGORITHM IS $O(n^2)$ IF YOU MULTIPLY TWO NUMBER OF n DIGITS EACH.

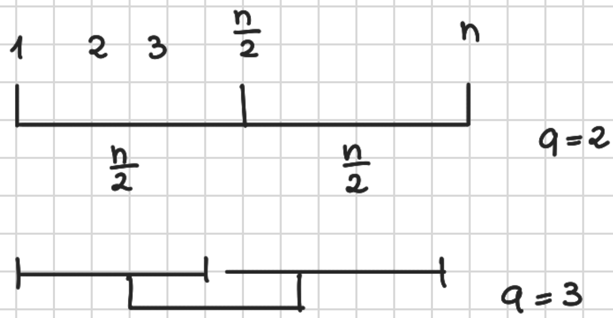
R2

$$\exists\, c>0, \quad T(n) \le \begin{cases} 2T\left(\frac{n}{2}\right)+cn & \forall n>2 \\ T(n)\le c & n\in\{0,1,2\} \end{cases}$$

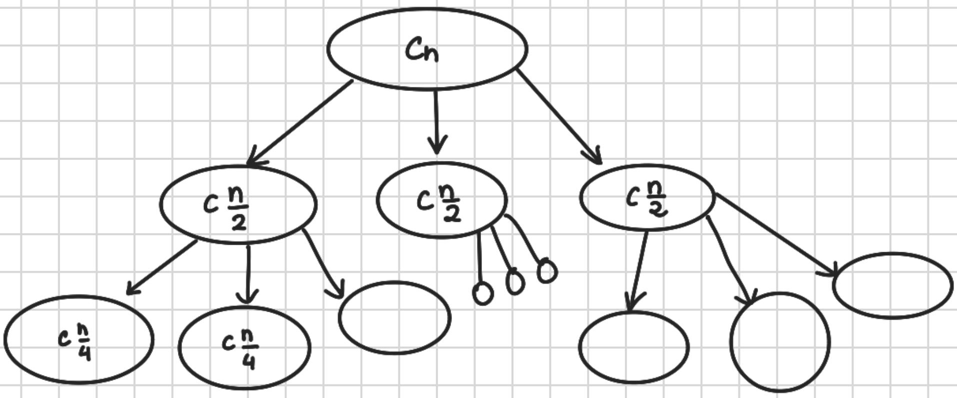
R2

$$\exists\, c>0, \quad T_q(n) \le \begin{cases} qT\left(\frac{n}{2}\right)+cn & \forall n>2 \\ c & \text{OTHERWISE} \end{cases}$$

$q \ge 3$



INPUT : n



It appears that in level i we have 3^i many calls and the runtime per call at level i is $c\cdot\frac{n}{2^i}$.

The proper cost at level i is $L_i \le q^i \cdot c\frac{n}{2^i} = \left(\frac{q}{2}\right)^i \cdot c\cdot n$

The total runtime

$$\sum_{i=0}^{\log n} L_i \le \sum_{i=0}^{\log n}\left(\left(\frac{q}{2}\right)^i\cdot cn\right)$$

Recall that $\sum_{i=0}^{t}\alpha^i = \alpha^0+\alpha^1+\alpha^2+\ldots+\alpha^t = \frac{\alpha^{t+1}-1}{\alpha-1}$

In our case, $\alpha = \frac{9}{2}$ and $t = \log_2 n$ thus,

$$\sum_{i=0}^{\log_2 n} L_i \leq cn \sum_{i=0}^{\log_2 n} \left(\frac{9}{2}\right)^i = cn \frac{\left(\frac{9}{2}\right)^{\log_2 n+1} - 1}{\frac{9}{2}-1} =$$

$$= 2cn \frac{\left(\frac{9}{2}\right)^{\log_2 n+1} - 1}{9-2}$$

$$= \frac{2cn}{9-2} \left(\frac{9}{2}\right)^{\log_2 n} \frac{9}{2}$$

$$(c^x)^y = n^{xy} = (c^y)^x$$

$$= \frac{2cn}{9-2} \left(2^{\log_2 \frac{9}{2}}\right)^{\log_2 n} \frac{9}{2}$$

$$= \frac{2cn}{9-2} \left(2^{\log_2 n}\right)^{\log_2 \frac{9}{2}} \frac{9}{2}$$

$$= \frac{2cn}{9-2} n^{\log_2 \frac{9}{2}} \frac{9}{2}$$

$$= \frac{2c}{9-2} n^{1+(\log_2(9)-1)} \frac{9}{2}$$

$$= \frac{9c}{9-2} n^{\log_2 9} = O(n^{\log_2 9})$$

C: The solution to R_9 is the $T_9(n) \leq O(n^{\log_2 9})$