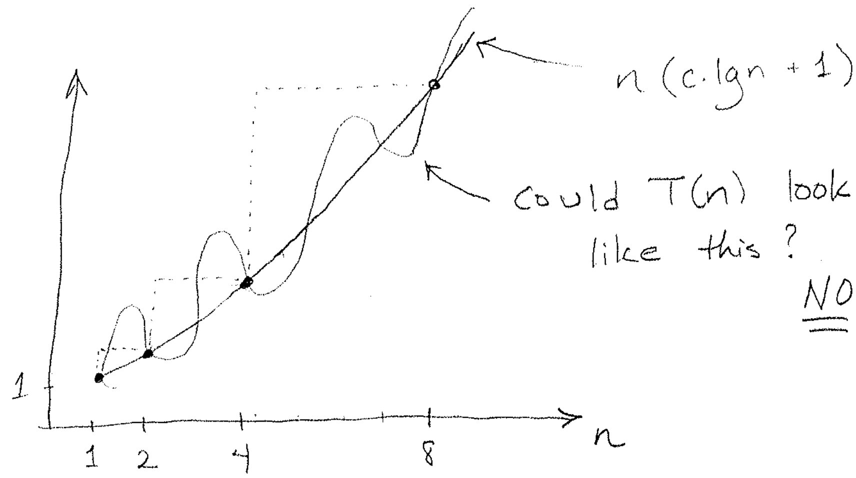
What about when $n \neq 2^k$?



1 $T(n)$ is an increasing function : $T(n) \leq T(n+1)$ for all $n \in \mathbb{N}$

Proof (by induction)

base : $T(1) = 1 \leq T(2) = 2 + 2c$ (for $c \geq 0$)

$$T(n+1) = T\left(\left\lceil \frac{n+1}{2} \right\rceil\right) + T\left(\left\lfloor \frac{n+1}{2} \right\rfloor\right) + c(n+1) \qquad c \geq 0$$

$n \geq \left\lceil \frac{n+1}{2} \right\rceil \geq \left\lceil \frac{n}{2} \right\rceil$ and I.H. $\longrightarrow \mathbin{\rotatebox{90}{$\geq$}}$ $\qquad$ $\mathbin{\rotatebox{90}{$\geq$}} \longleftarrow$ $n \geq \left\lfloor \frac{n+1}{2} \right\rfloor \geq \left\lfloor \frac{n}{2} \right\rfloor$ and I. H

$$T(n+1) \geq T\left(\left\lceil \frac{n}{2} \right\rceil\right) + T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + cn$$

$$= T(n)$$

because $2^{\lceil \lg n \rceil}$ is a power of 2 so out previous exact analysis holds.

$$1 \Rightarrow \quad T(n) \leq T(\underbrace{2^{\lceil \lg n \rceil}}_{\text{smallest power of 2 that is } \geq n}) = 2^{\lceil \lg n \rceil}\left(c \lceil \lg n \rceil + 1\right)$$

$$< 2^{(\lg n + 1)}\left(c(\lg n + 1) + 1\right)$$

$$= 2n\left(c \lg n + c + 1\right)$$

$$= 2c\, n \lg n + 2(c+1)\, n$$

$$\leq d\, n \lg n \quad \text{for } n \geq 2$$

$$d = 4c + 2$$

Pitfalls

(A) Let $T(n) = T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + n$, $T(1) = 1$

$\boxed{\text{Wrong}}$ Guess: $T(n) \in O(n)$ i.e. $T(n) \le c \cdot n$ for some $c > 0$ and all $n \ge n_0$

Verify by induction:

Inductive Step
$$T(n) = T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + n$$
$$\le c \lceil n/2 \rceil + c \lfloor n/2 \rfloor + n \quad \text{by Ind. Hyp}$$

Note: $\lceil \frac{n}{2} \rceil + \lfloor \frac{n}{2} \rfloor = n$ for all $n \in \mathbb{N}$

$$= cn + n$$
$$= (c+1)n$$
$$\le d \cdot n \quad \text{for } d \ge c+1 \quad \text{so } T(n) \in O(n)$$
$$n \ge n_0$$

Error! We did not prove the exact inductive hypothesis that $T(n) \le \underline{c} \cdot n$

We cannot change c.

(B) Let $T(n) = T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + 1$, $T(1) = 1$

Guess $T(n) \le c \cdot n$

Verify by induction
$$T(n) = T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + 1$$
$$\le c\lceil \tfrac{n}{2} \rceil + c \lfloor n/2 \rfloor + 1$$
$$= cn + 1$$

oops!

Sometimes a stronger inductive hypothesis works.

Guess $T(n) \le c \cdot n - 1$
$$T(n) = T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + 1$$
$$\le c\lceil n/2 \rceil - 1 + c\lfloor n/2 \rfloor - 1 + 1$$
$$= cn - 1$$

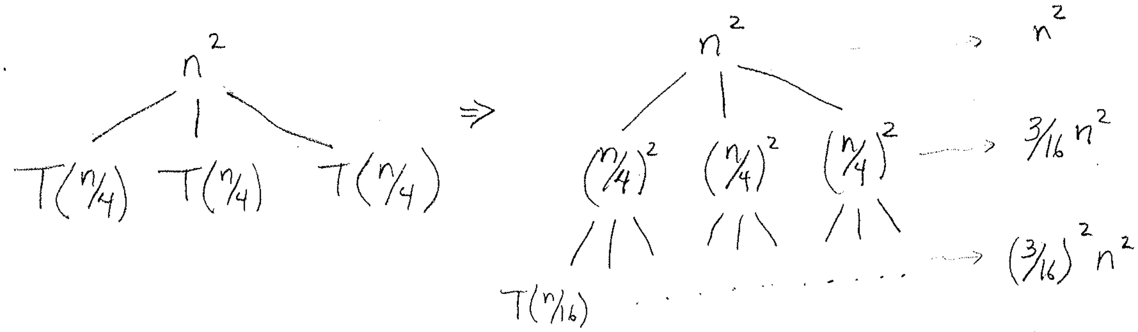
Base case $T(1) = 1 \le c \cdot 1 - 1$ for $c \ge 2$

Recursion Tree method

- Use a tree to picture the expansion of a recurrence relation
    - record the nonrecursive contribution (cost) at a node
    - record the recursive contribution(s) in the node's subtrees

Example $T(n) = 3T(n/4) + n^2$, $T(1) = 1$

Assume n is a power of 4.



- Calculate the cost done on each level

i^{th} level $\rightarrow (3/16)^i n^2$

- Add up all cost on all levels

$$\text{Total} = \sum_{i=0}^{\log_4 n - 1} (3/16)^i n^2 + \underbrace{3^{\log_4 n}}_{\substack{\text{leaves} \\ T(1)=1}}$$

$$< \sum_{i=0}^{\infty} (3/16)^i n^2 + 3^{\log_4 n}$$

$$= n^2\left(\frac{1}{1 - 3/16}\right) + 3^{\log_4 n}$$

$$= 16/13\, n^2 + 3^{\log_4 n} = \frac{16}{13} n^2 + n^{\log_4 3}$$

$$x^{\log_b y} = y^{\log_b x}$$

since

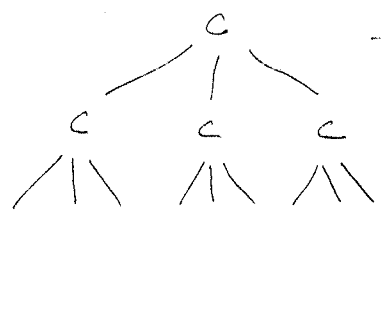
$$\log_b\left(x^{\log_b y}\right) = \log_b\left(y^{\log_b x}\right)$$

$$\log_b y \log_b x = \log_b x \log_b y$$

$\Rightarrow T(n) \in O(n^2)$

- Verify using induction [if you're not sure]

Example 2 $T(n) \le 3T(\lfloor n/2 \rfloor) + c$, $T(1) \le d$



c ← Assume n is a power of 2

$3c$

$3^2 c$

number of levels before we reach leaves → $\overline{\lfloor \lg n \rfloor - 1}$; cost at leaves ↘

$$= \sum_{i=0}^{\lfloor \lg n \rfloor - 1} 3^i \cdot c + d \cdot 3^{\lfloor \lg n \rfloor}$$

$$= c \cdot \frac{3^{\lfloor \lg n \rfloor} - 1}{3 - 1} + d \cdot 3^{\lfloor \lg n \rfloor}$$

$$< c\, 3^{\lg n} + d \cdot 3^{\lg n}$$

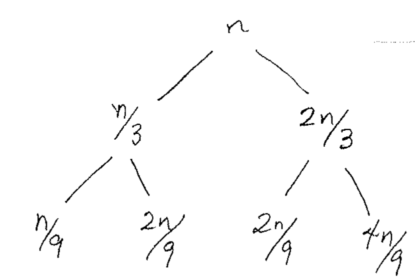
$$= (c+d)\, n^{\lg 3} \in O(n^{\lg 3})$$

$\lg 3 \approx 1.58\ldots$

Remember

$$\sum_{i=0}^{k} a^i = \frac{a^{k+1} - 1}{a - 1}$$

Example 3 $T(n) = T(\lfloor n/3 \rfloor) + T(\lfloor 2n/3 \rfloor) + n$, $T(1) = 1$



→ n

→ n

→ n

Let's suppose floor $\lfloor\ \rfloor$ doesn't matter

NOTE: Tree is not balanced (ie some root-to-leaf paths are longer than others)

max number of levels before we reach leaves → $\overline{\log_{3/2} n - 1}$

$$\le \sum_{i=0}^{\log_{3/2} n - 1} n + \text{cost at leaves}$$

$$\le n \log_{3/2} n +$$

$$\le c \cdot n \log n$$

← this is our guess

cost at leaves $\le$ total cost at internal nodes of tree

cost at leaves $\le 2^{\log_{3/2} n} = n^{\log_{3/2} 2}$

$= n^{1.7\ldots}$

That's too big

Verifying $T(n) \in O(n \log n)$

i.e. $T(n) \le c \cdot n \log_3 n$ for some c and $n \ge n_0$

choosing base 3 is convenient

Ind. Step

$$T(n) = T(\lfloor n/3 \rfloor) + T(\lfloor \tfrac{2n}{3} \rfloor) + n$$

$$\le c \lfloor n/3 \rfloor \log_3 \lfloor n/3 \rfloor + c \lfloor \tfrac{2n}{3} \rfloor \log_3 \lfloor \tfrac{2n}{3} \rfloor + n$$

$$\le c\,(n/3) \log_3 (n/3) + c\left(\tfrac{2n}{3}\right) \log_3 \left(\tfrac{2n}{3}\right) + n$$

$$\le c\, n/3 \,(\log_3 n - 1) + c\left(\tfrac{2n}{3}\right)\left(\log_3 n - \log_3 3/2\right) + n$$

$$\le c\, n/3 \log_3 n - c\,n/3 + c\, \tfrac{2n}{3} \log_3 n + n$$

$$= c\, n \log_3 n - c \cdot n/3 + n \qquad \text{true for } c \ge 3$$

Base case then holds for $n \ge n_0 = 3$