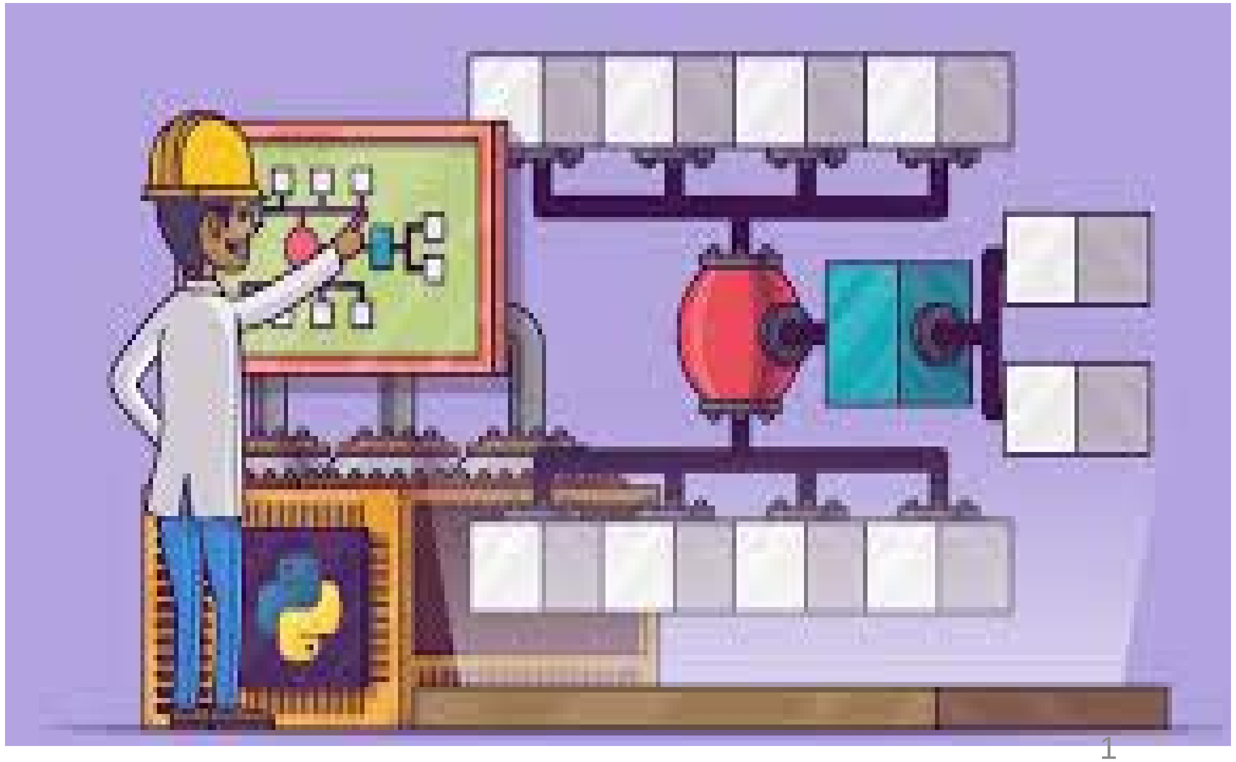
ساختمان داده ها

مدرس:
سمانه حسینی سمنانی

دانشگاه صنعتی اصفهان- دانشکده برق و کامپیوتر



Recursion Complexity

- Merge sort recursion complexity function:

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1 \,, \\ 2T(n/2) + \Theta(n) & \text{if } n > 1 \,, \end{cases}$$

- Different methods to solve such recurrences:
  - Substitution method
  - Recursion-tree method
  - Master method

سمانه حسینی سمنانی
هیات علمی دانشکده برق و کامپیوتر- دانشگاه صنعتی اصفهان

Substitution method

- we guess a bound and then use mathematical induction to prove our guess correct.

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1 , \\ 2T(n/2) + \Theta(n) & \text{if } n > 1 , \end{cases}$$

- Guess 1: $\forall n \quad T(n) < 4n$
- $For\ n = 1 \ T(1) = 1 < 4 * 1$
- فرض: $\forall k < n \quad T(n) < 4n$
- حکم: $k = n \quad T(n) \leq 4n$

Substitution method

- Guess 1: $\forall n \quad T(n) < cn$
- فرض: $\forall k < n \quad T(n) < cn$
- حکم: $k = n \quad T(n) \leq cn$

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1 , \\ 2T(n/2) + \Theta(n) & \text{if } n > 1 , \end{cases}$$

Substitution method: change the guess to n^2

- Guess 1: $\forall n \quad T(n) < cn^2$
- فرض: $\forall k < n \quad T(n) < cn^2$
- حکم: $k = n \quad T(n) \le cn^2$

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1, \\ 2T(n/2) + \Theta(n) & \text{if } n > 1, \end{cases}$$

Substitution method: guess is nlogn

- $T(n) = O(nlogn)$
- Guess 1: $\forall n \quad T(n) < cnlogn$
- فرض: $\forall k < n \quad T(n) < cnlogn$
- حکم: $k = n \quad T(n) \leq cnlogn$

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1 \,, \\ 2T(n/2) + \Theta(n) & \text{if } n > 1 \,, \end{cases}$$

$$T(n) = 2T(\lfloor n/2 \rfloor) + n$$

$$\begin{aligned} T(n) &\leq 2(c\lfloor n/2 \rfloor \lg(\lfloor n/2 \rfloor)) + n \\ &\leq cn \lg(n/2) + n \\ &= cn \lg n - cn \lg 2 + n \\ &= cn \lg n - cn + n \\ &\leq cn \lg n \,, \end{aligned}$$

- $For\ n = 1,\ T(1) = 1\ \therefore\ T(1) \leq c1 \lg 1 = 0.$

last step holds as long as $c \geq 1$.

Substitution method: guess is nlogn

- *For* $n = 1$, $T(1) = 1 \therefore \; T(1) \le c1 \lg 1 = 0$.
- $T(n) \le cn \lg n$ for $n \ge n_0$
- $T(2) = 4$ and $T(3) = 5$.
- $n_0 = 2$. , $c \ge 2$

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1 \,, \\ 2T(n/2) + \Theta(n) & \text{if } n > 1 \,, \end{cases}$$

Substitution method: guess is nlogn

- $T(n) = \Omega(nlogn)$
- Guess 1: $\forall n \quad T(n) \geq cnlogn$
- فرض: $\forall k < n \quad T(n) > cnlogn$
- حکم: $k = n \quad T(n) \geq cnlogn$

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1 , \\ 2T(n/2) + \Theta(n) & \text{if } n > 1 , \end{cases}$$

- $C = 1 \rightarrow T(n) = O(nlogn), T(n) = \Omega(nlogn) \rightarrow T(n) = \theta(nlogn)$

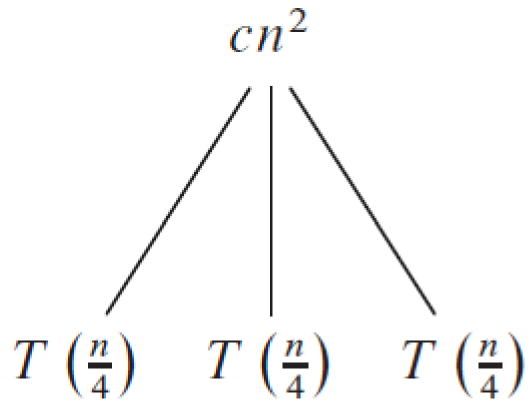
Recursion-tree Example

- Problem of previous method: hard to find good guess.
- Recursion tree is a good method to find a good guess
- That guess later can be proved using Substitution method

$$T(n) = 3T(\lfloor n/4 \rfloor) + \Theta(n^2)$$

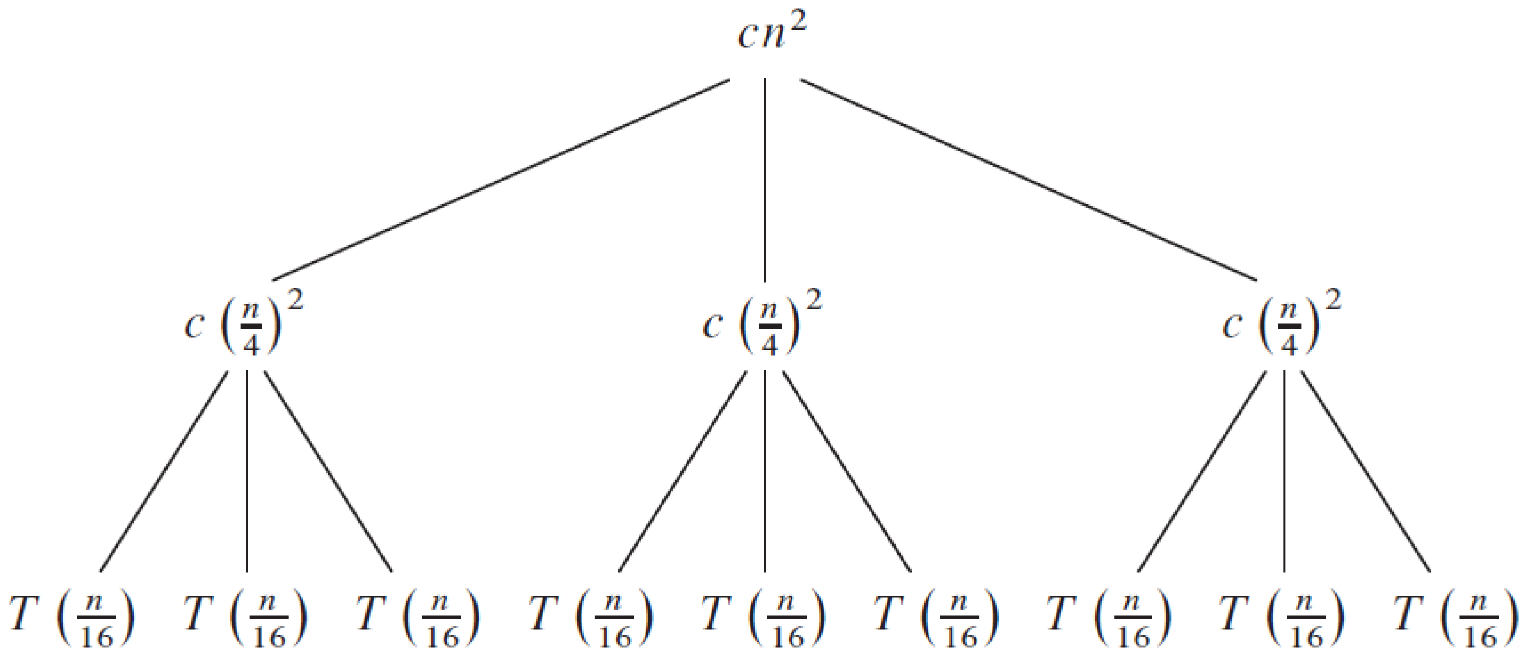
Recursion-tree Example

- $T(n) = 3T(\lfloor n/4 \rfloor) + \Theta(n^2)$
- $T(n) = 3T(n/4) + cn^2$
- we assume that n is an exact power of 4



Recursion-tree Example

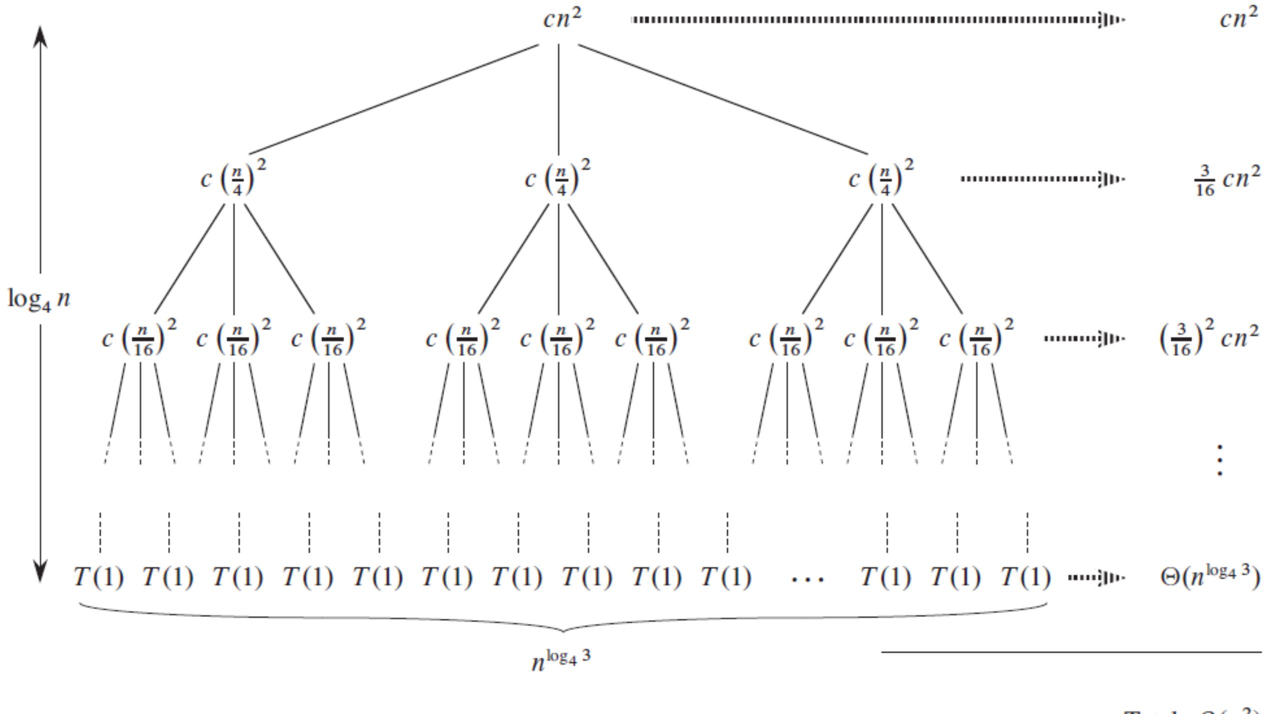
- $T(n) = 3T(n/4) + cn^2$



Recursion-tree Example

- $T(n) = 3T(n/4) + cn^2$
- Problem size at level $i = \frac{n}{4^i}$
- At last level $1 = \frac{n}{4^i}$
- $i = \log_4 n$
- tree has $\log_4 n + 1$ levels

  (at depths $0, 1, 2, \ldots, \log_4 n$)

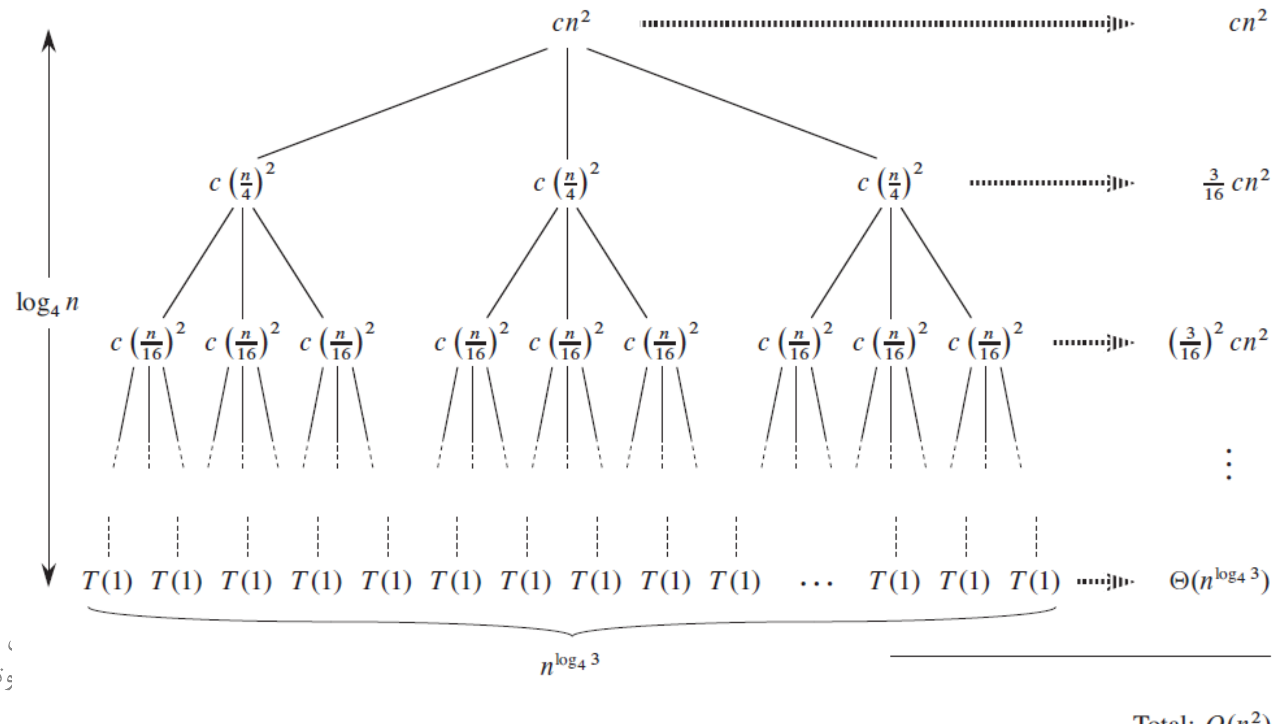


(d)

سمانه حسینی سمنانی
هیات علمی دانشکده برق و کامپیوتر- دانشگاه صنعتی اصفهان

Recursion-tree Example

- Each level has three times more nodes than the level above
- the number of nodes at depth i is 3^i .
- The bottom level, at depth $\log_4 n$ has $3^{\log_4 n} = n^{\log_4 3}$ nodes

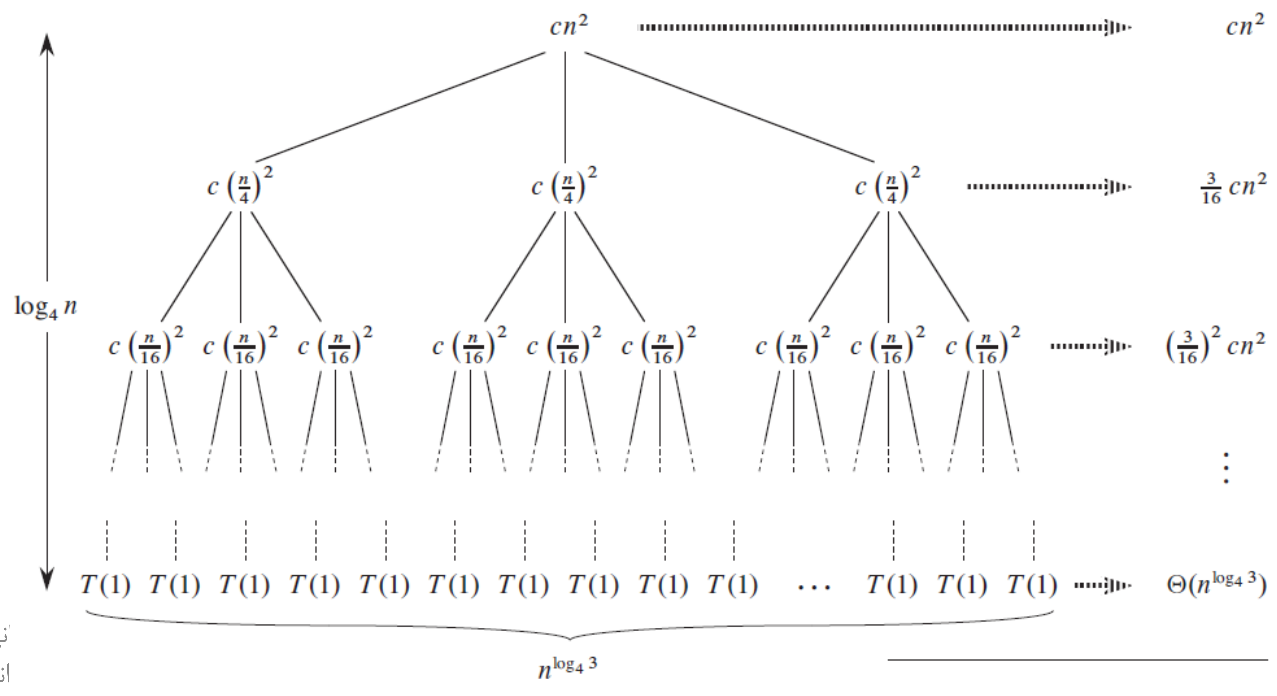


(d)

Recursion-tree Example

$$
\begin{aligned}
T(n) &= cn^2 + \frac{3}{16}cn^2 + \left(\frac{3}{16}\right)^2 cn^2 + \cdots + \left(\frac{3}{16}\right)^{\log_4 n-1} cn^2 + \Theta(n^{\log_4 3}) \\
&= \sum_{i=0}^{\log_4 n-1} \left(\frac{3}{16}\right)^i cn^2 + \Theta(n^{\log_4 3}) \\
&= \frac{(3/16)^{\log_4 n} - 1}{(3/16) - 1} cn^2 + \Theta(n^{\log_4 3})
\end{aligned}
$$

$$
\textit{if} \quad |x| < 1, \quad \textit{then} \quad \sum_{k=0}^{\infty} x^k = \frac{1}{1-x}
$$

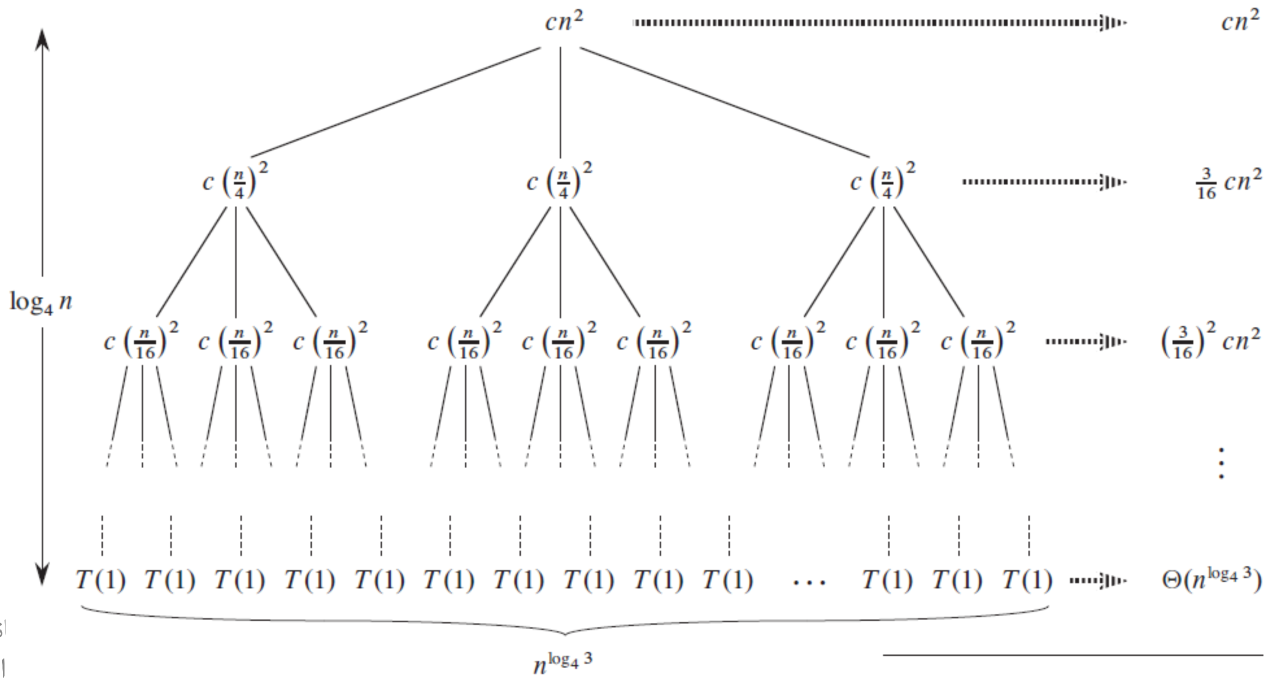


(d)

انی
انشگاه صنعتی اصفهان

Recursion-tree Example

$$
\begin{aligned}
T(n) &= \sum_{i=0}^{\log_4 n - 1} \left(\frac{3}{16}\right)^i cn^2 + \Theta(n^{\log_4 3}) \\
&< \sum_{i=0}^{\infty} \left(\frac{3}{16}\right)^i cn^2 + \Theta(n^{\log_4 3}) \\
&= \frac{1}{1-(3/16)} cn^2 + \Theta(n^{\log_4 3}) \\
&= \frac{16}{13} cn^2 + \Theta(n^{\log_4 3}) \\
&= O(n^2) .
\end{aligned}
$$



(d)

Total: $O(n^2)$

Prove the guess using Substitution

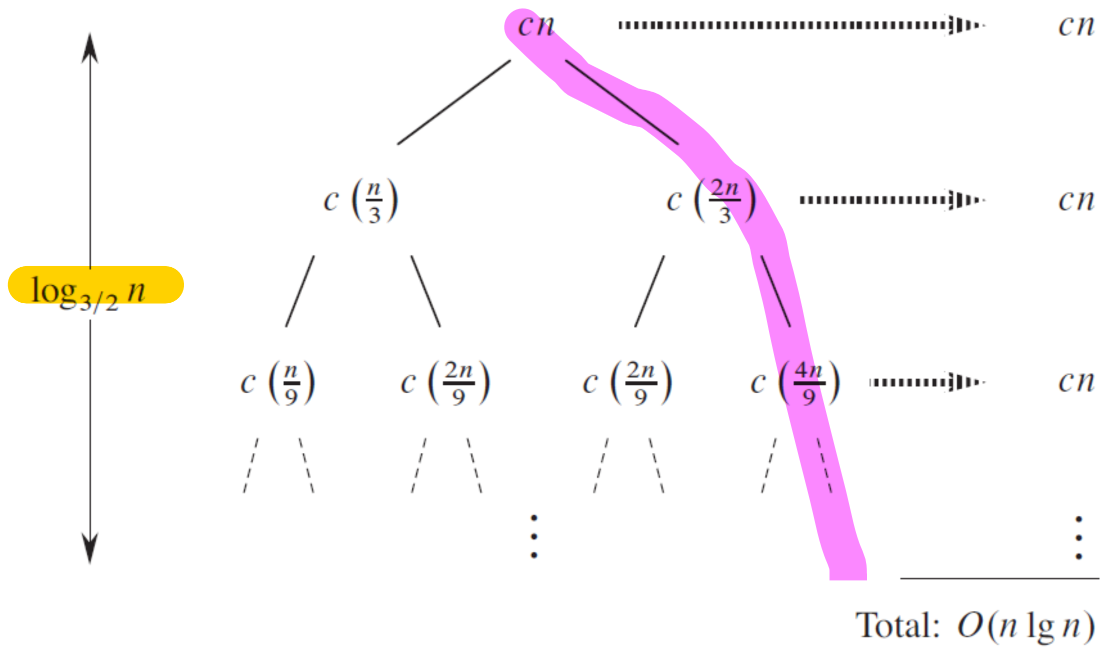
$$T(n) = 3\,T(\lfloor n/4 \rfloor) + c\,n^2$$

- Guess: $T(n) = O(n^2)$
- We want to show that $T(n) \le dn^2$ for some constant $d > 0$.

$$\begin{aligned} T(n) &\le 3T(\lfloor n/4 \rfloor) + cn^2 \\ &\le 3d\,\lfloor n/4 \rfloor^2 + cn^2 \\ &\le 3d(n/4)^2 + cn^2 \\ &= \frac{3}{16}dn^2 + cn^2 \\ &\le dn^2 , \qquad\qquad d \ge (16/13)c \end{aligned}$$

Recursion-tree Example

$$T(n) = T(n/3) + T(2n/3) + O(n)$$



سمانه حسینی سمنانی
هیات علمی دانشکده برق و کامپیوتر- دانشگاه صنعتی اصفهان

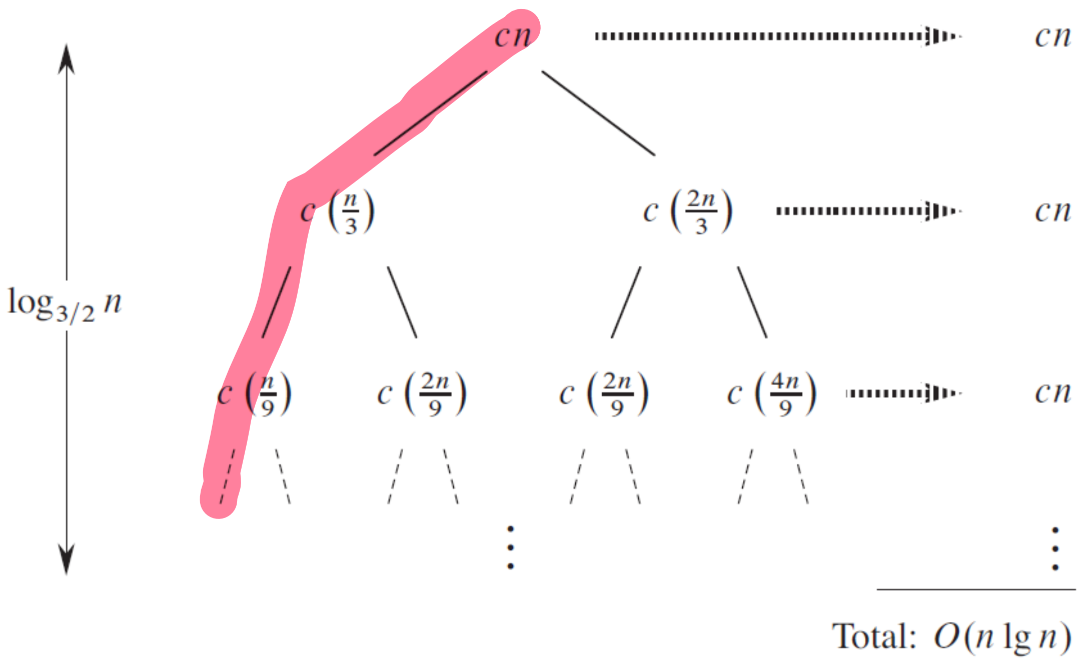
Recursion-tree Example

$$n \rightarrow (2/3)n \rightarrow (2/3)^2 n \rightarrow \cdots \rightarrow 1.$$

Since $(2/3)^k n = 1$ when $k = \log_{3/2} n$,

$$O(cn \log_{3/2} n) = O(n \lg n).$$

We can prove this guess using Substitution



A new type of recursion

$$T(n) = 2T(\sqrt{n}) + logn$$

$$m = logn \rightarrow T(2^m) = 2T\left(2^{\frac{m}{2}}\right) + m$$

$$s(m) = T(2^m) \rightarrow s(m) = 2s\left(\frac{m}{2}\right) + m$$

$$s(m) = \theta(mlogm)$$

$$T(2^m) = \theta(mlogm)$$

$$T(n) = \theta(logn\ loglogn)$$