Recurrences

Technicalities

- $T(n)=\Theta(1)$ for sufficiently small n
- $T(\lfloor n/b \rfloor)$ and $T(\lceil n/b \rceil)$ are often denoted as $T(n/b)$.
    - In most cases, it does not matter!
    - But it matters in exam.
- We mainly consider monotonically increasing $T(n)$, i.e., $T(n') \le T(n)$ if $n' \le n$.

Solving Recurrences

- 3 methods in the textbook
  - The substitution method
    - Guess the answer + induction
  - The recursion tree method
    - Iteration method
  - The master method

Substitution Method

- Step 1: Guess the form of the solution
- Step 2: Use mathematical induction
  - Find out constants c and n_0.
  - Prove that the solution works

Example: Merge Sort

- $T(n)=T(\lfloor n/2 \rfloor)+T(\lceil n/2 \rceil)+n$
- Guess: T(n) should be O(nlogn)
- Goal: Find out c and n_0.
- Assume $T(n) \le cn\log n$

Find c, n_0 later

- Induction basis: $T(n_0) \le cn_0 \log n_0$
- Induction hypothesis: $T(k) \le ck\log k$ for $n_0 \le k < n$

Example: Merge Sort

$\boxed{\text{Goal: } T(n) \le cn\log n}$

- Inductive step:
  $T(n)=T(\lfloor n/2 \rfloor)+T(\lceil n/2 \rceil)+n$
  $\le c\lfloor n/2 \rfloor \log\lfloor n/2 \rfloor + c\lceil n/2 \rceil \log\lceil n/2 \rceil + n$
  $\le c(\lfloor n/2 \rfloor + \lceil n/2 \rceil)\log(n/1.5)+n$
  $\le cn\log n - cn\log(1.5)+n$
  $\le cn\log n$ .... true for $1 \le c\log(1.5)$ & $n \ge 2$
- So we should pick $c \ge 1/\log(1.5)$ & $n \ge 2$.
- Pick $c=\max(1/\log(1.5), T(10)/10)$, $n_0=10$, we can verify that $T(n_0) \le cn_0\log n_0$.

Subtleties

- $T(n)=T(\lfloor n/2 \rfloor)+T(\lceil n/2 \rceil)+1$
- Guess: $T(n)$ should be $O(n)$
- Goal: Find out c and n_0.
- Assume $T(n) \leq cn$
- Induction basis: $T(n_0) \leq cn_0$
- Induction hypothesis: $T(k) \leq ck$ for $n_0 \leq k < n$

Subtleties

$\boxed{\text{Goal: } T(n) \le cn}$

- Inductive step:
  $T(n)=T(\lfloor n/2 \rfloor)+T(\lceil n/2 \rceil)+1$
  $\le c\lfloor n/2 \rfloor+c\lceil n/2 \rceil+1$
  $=cn+1$ .... does not work!
- So we should try to assume $T(n) \le cn-b$
- Induction basis: $T(n_0) \le cn_0-b$
- Induction hypothesis: $T(k) \le ck-b$ for $n_0 \le k < n$

Subtleties

$\boxed{\text{Goal: } T(n) \le cn - b}$

- Inductive step:
  $T(n) = T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + 1$
  $\le c\lfloor n/2 \rfloor - b + c\lceil n/2 \rceil - b + 1$
  $= cn - 2b + 1$
  $\le cn - b$ .... true for $b \ge 1$
- Pick $c = (T(10)+1)/10$, $n_0 = 10$, $b = 1$ we can verify that $T(n_0) \le cn_0 - b$.

Avoiding Pitfall

Note: wrong guess

- $T(n)=2T(n/2)+n$
- Guess: $T(n)$ should be $O(n)$
- Goal: Find out c and n_0.
- Assume $T(n) \le cn$
- Induction basis: $T(n_0) \le cn_0$
- Induction hypothesis: $T(k) \le ck$ for $n_0 \le k < n$

Avoiding Pitfall

- Inductive step:
  $T(n)=2T(n/2)+n$
  $\leq cn/2+cn/2+n$
  $=cn+n$
  $=O(n)$ .... WRONG!!!
- Does not match the hypothesis!
- c should be identical for all n.

Changing Variable

- $T(n)=2T(n^{0.5})+\lg n$ take $n=2^m$
- $T(2^m)=2T(2^{m/2})+m$ rename $S(m)=T(2^m)$
- $S(m)=2S(m/2)+m$
- We have $S(m)=O(m\log m)$
- $T(n)=T(2^m)=O(m\log m)=O(\log n \log\log n)$

Recursion Tree

- Iteration method
- Example 1:

$T(n)=2T(n/3)+n=2^2T(n/9)+2n/3+n$

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

$\boxed{\log_3 n=\log n/\log 3}$

$=2^{\log n/\log 3}\Theta(1)+\ldots+(2/3)^2n+(2/3)n+n$

$=n^{\log 2/\log 3}\Theta(1)+\ldots+(2/3)^2n+(2/3)n+n$

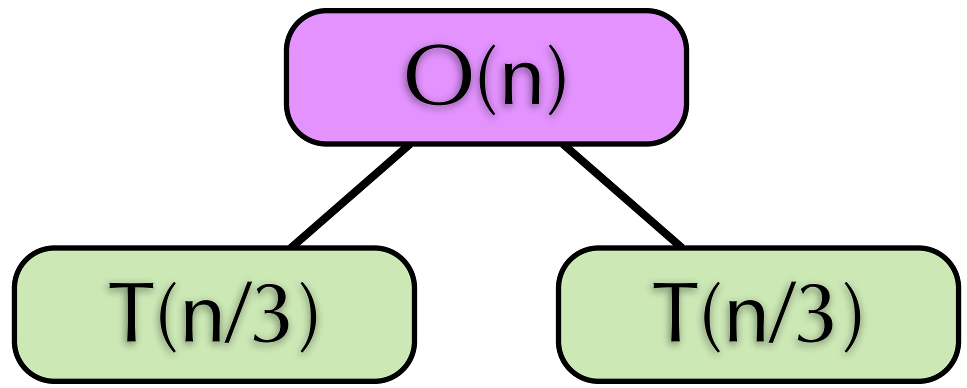
$=\Theta(n^{\log_3 2})+n(1+(2/3)+(2/3)^2+\ldots)$

$\leq o(n)+3n=O(n)$

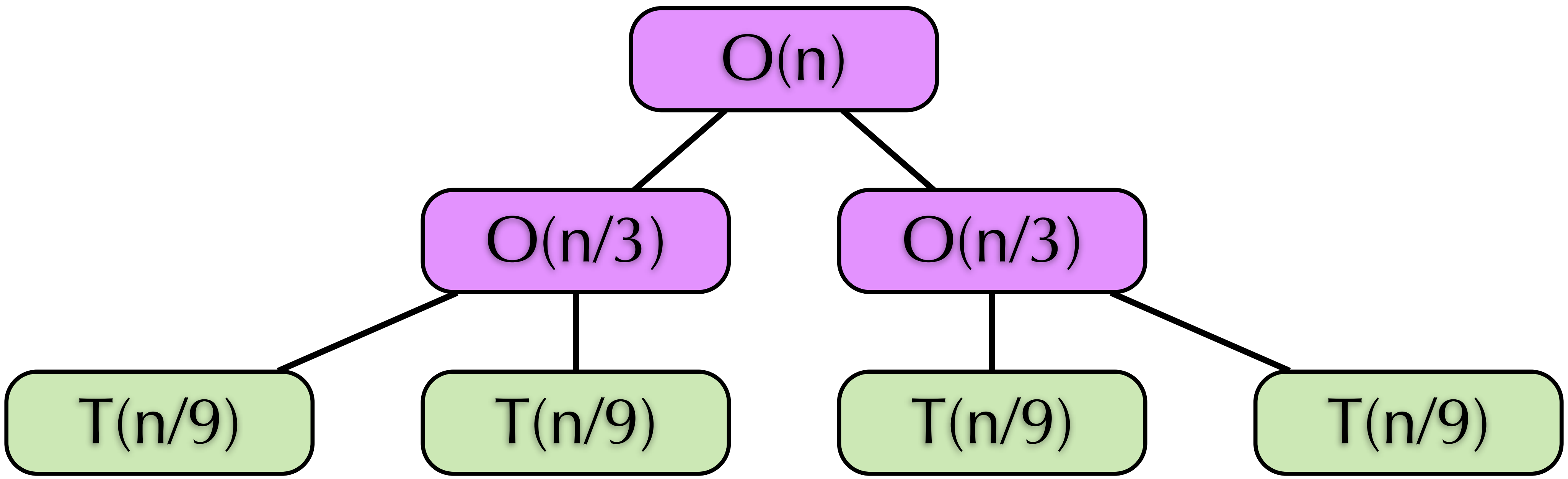
$$T(n)=2T(n/3)+O(n)$$

$T(n)$

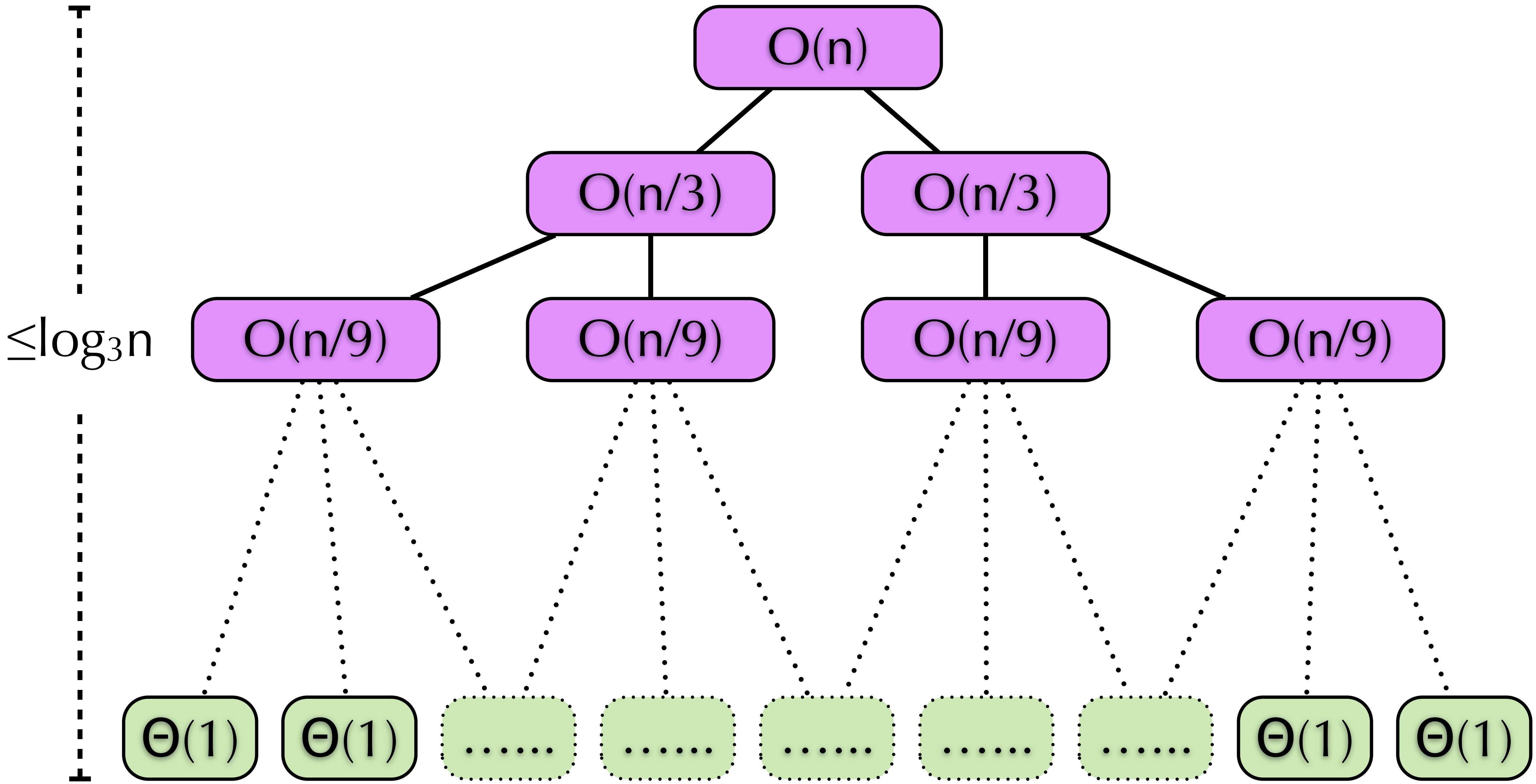
$$T(n)=2T(n/3)+O(n)$$



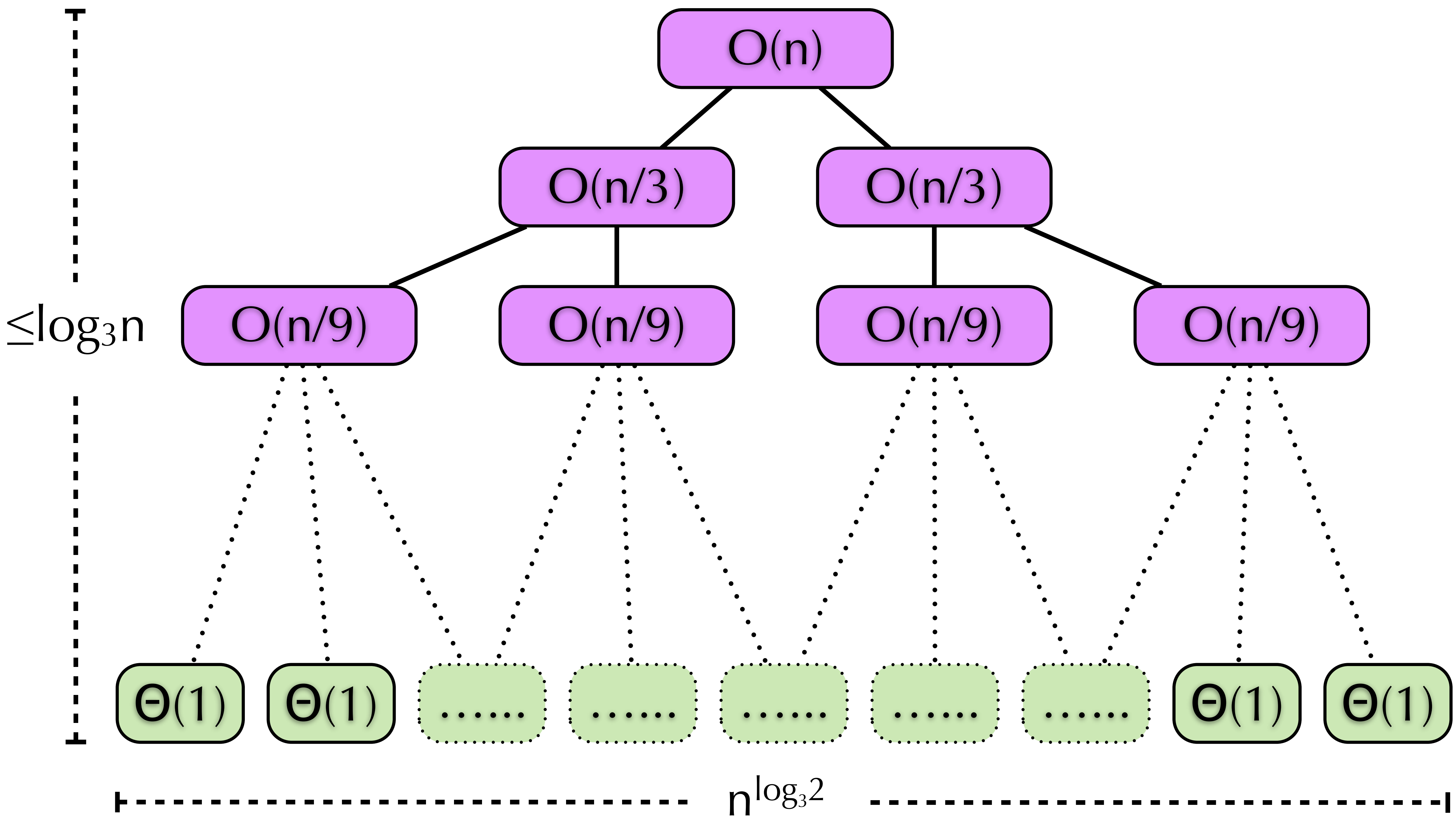
$T(n)=2T(n/3)+O(n)$



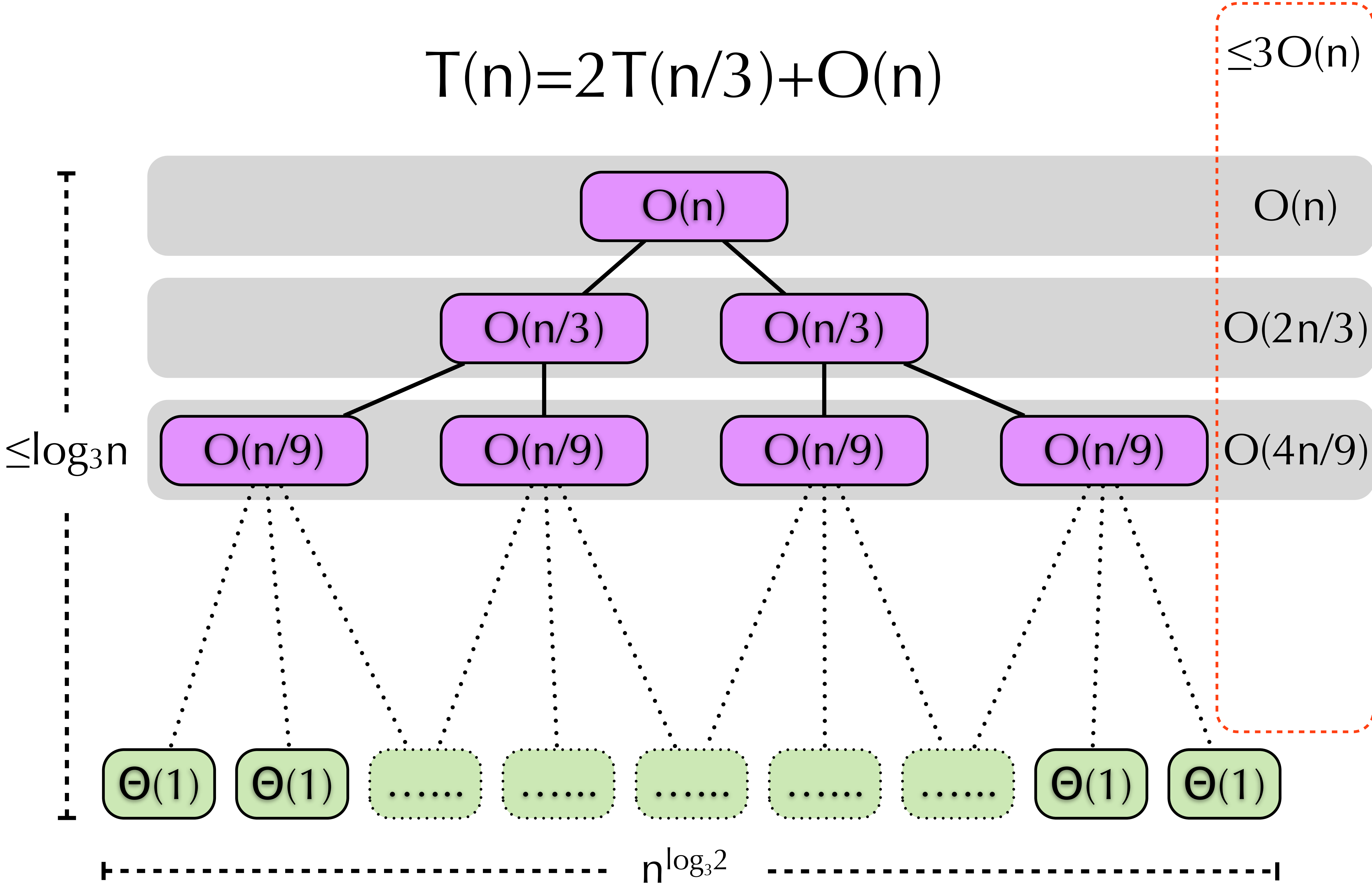
$$T(n)=2T(n/3)+O(n)$$



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



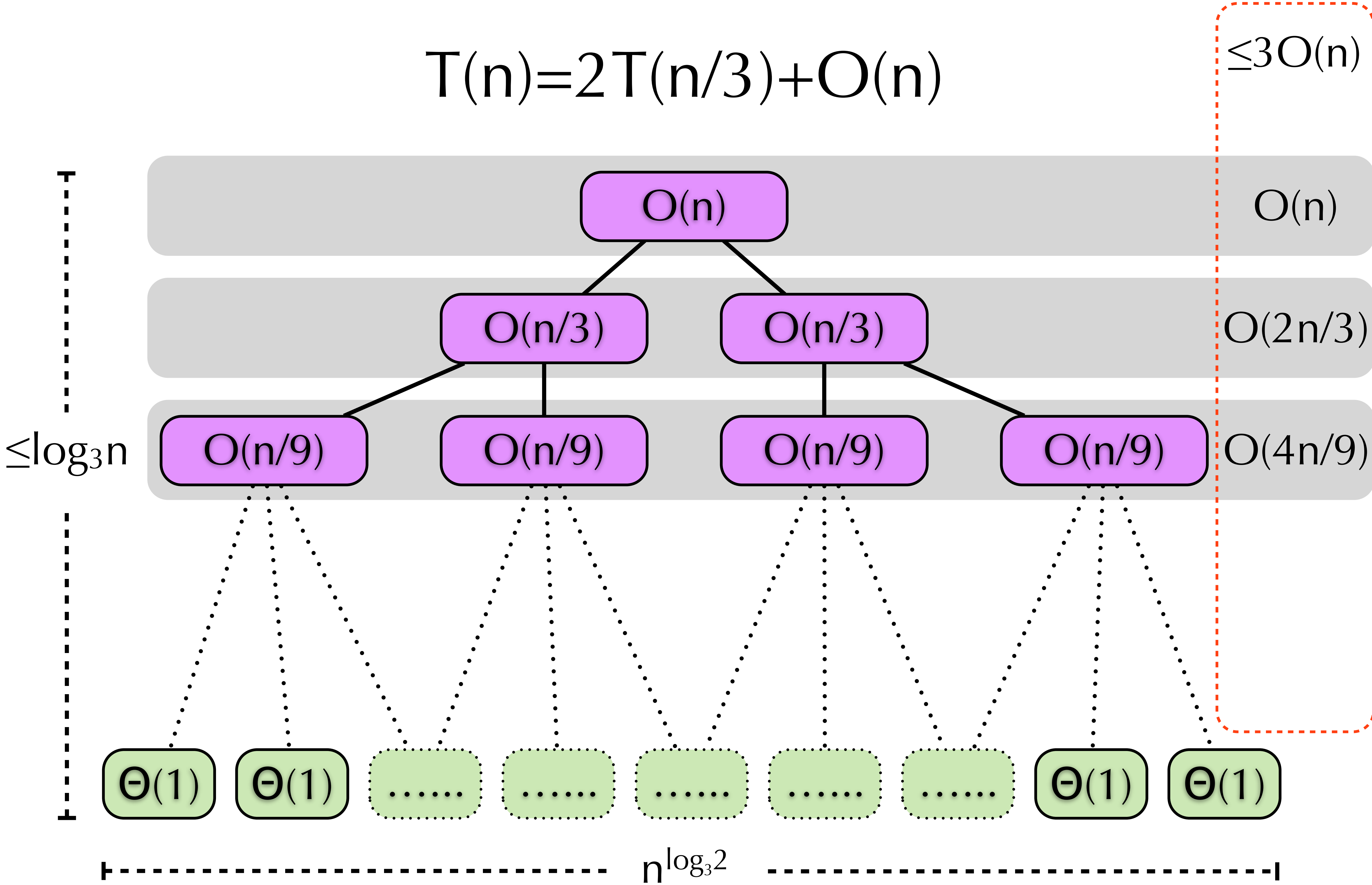
$$T(n)=2T(n/3)+O(n)$$



≤3O(n)
O(n)
O(n/3)
O(n/3)
O(2n/3)
≤log3n
O(n/9)
O(n/9)
O(n/9)
O(n/9)
O(4n/9)
Θ(1)
Θ(1)
……
……
……
……
……
Θ(1)
Θ(1)
n^log3 2

Dominating!

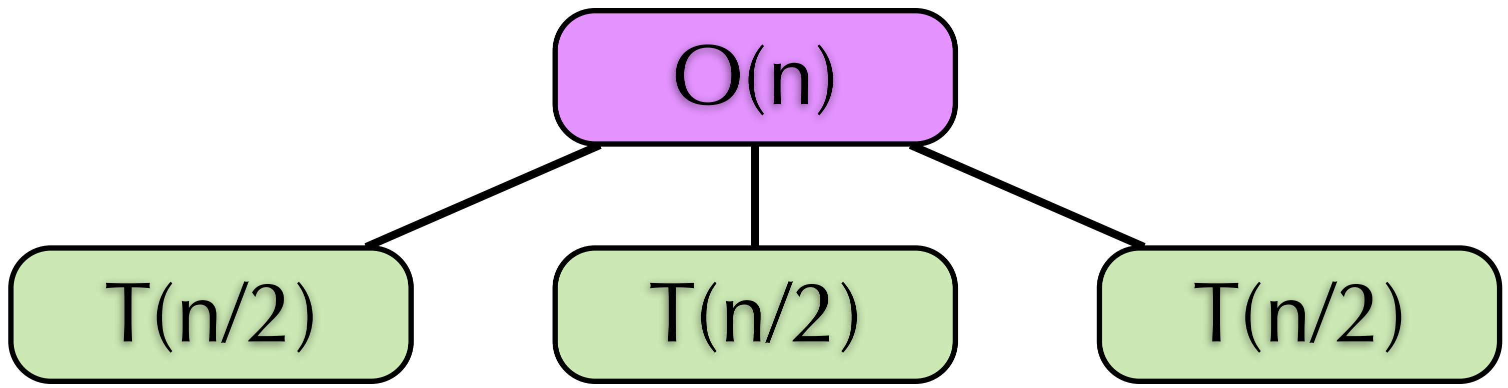
$$T(n)=2T(n/3)+O(n)$$



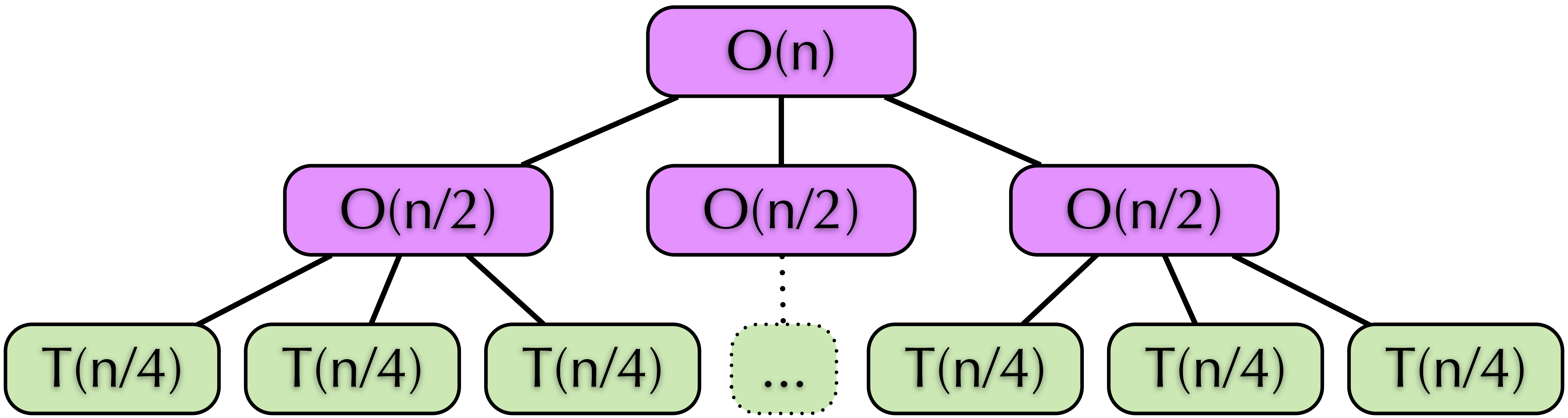
$$T(n)=3T(n/2)+O(n)$$

$T(n)$

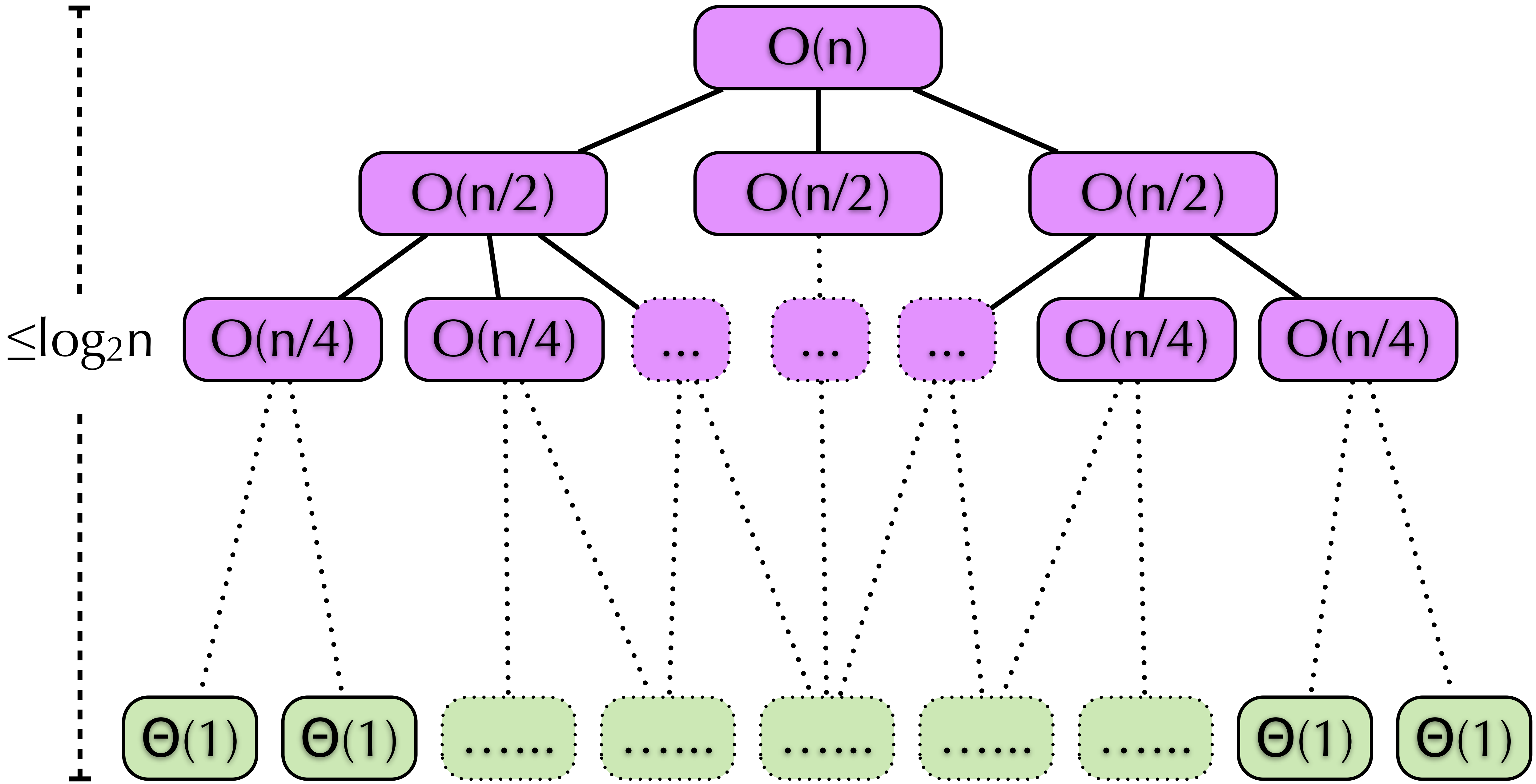
$$T(n)=3T(n/2)+O(n)$$



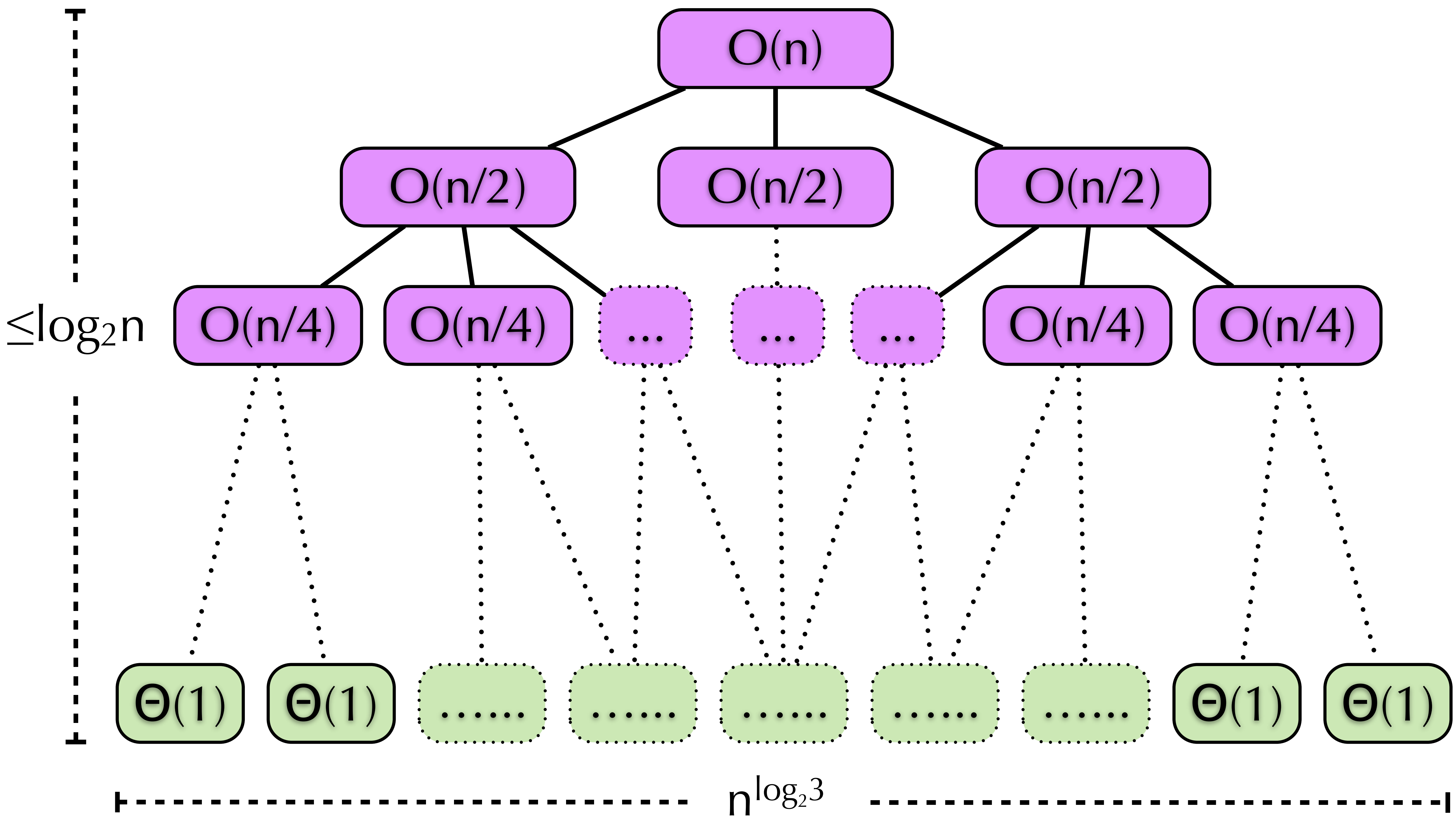
$$T(n)=3T(n/2)+O(n)$$



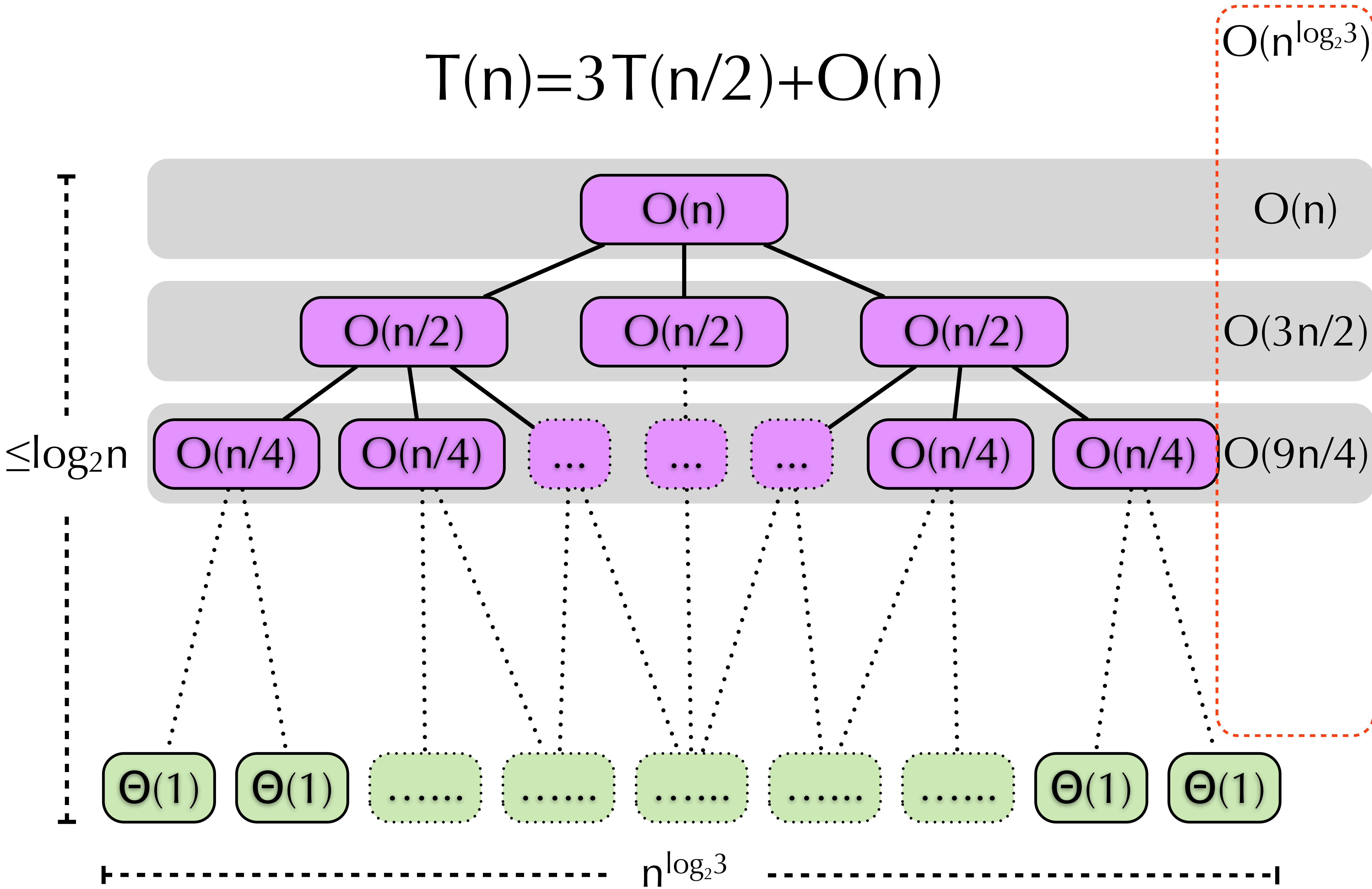
$T(n)=3T(n/2)+O(n)$



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



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



O(n^log₂3)
O(n)
O(n/2)
O(n/2)
O(n/2)
O(3n/2)
≤log₂n
O(n/4)
O(n/4)
...
...
...
O(n/4)
O(n/4)
O(9n/4)
Θ(1)
Θ(1)
......
......
......
......
......
Θ(1)
Θ(1)
n^log₂3

The Master Theorem

- $T(n)=aT(n/b)+f(n)$
  - $a\geq 1$ and $b>0$ are constant.
  - n/b can be $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$.
- $T(n)=\Theta(n^{\log_b a})$ if $f(n)=O(n^{\log_b a-\varepsilon})$ for some constant $\varepsilon>0$.
- $T(n)=\Theta(n^{\log_b a}\log_b n)$ if $f(n)=\Theta(n^{\log_b a})$.
- $T(n)=\Theta(f(n))$ if $f(n)=\Omega(n^{\log_b a+\varepsilon})$ for some constant $\varepsilon>0$ and $af(n/b)\leq cf(n)$ for some constant $c<1$ and all sufficiently large n.

Examples

- $T(n)=7T(n/2)+\Theta(n^2)$ $\Theta(n^2)=O(n^{\log_2 7-\varepsilon})$
  - $T(n)=\Theta(n^{\log_2 7})$
- $T(n)=T(n/3)+1$ $1=\Theta(n^{\log_3 1})=\Theta(1)$
  - $T(n)=\Theta(\log_3 n)$
- $T(n)=2T(n/3)+n\log n$ $n\log n=\Omega(n^{\log_3 2+\varepsilon})$
  $(2n/3)(\log n-\log 3)\leq(2/3)n\log n$
  - $T(n)=\Theta(n\log n)$

Remarks

- Not in the 3 cases: $T(n)=2T(n/2)+n\log n$
- $\log n=o(n^{\varepsilon})$ for any constant $\varepsilon>0$.
- Due to Exercise 4.6-2 in the textbook, we have $T(n)=\Theta(n\log^2 n)$.
- Exercise 4.6-2: $T(n)=\Theta(n^{\log_b a}\log^{k+1} n)$ if $k\geq 0$ and $f(n)=\Theta(n^{\log_b a}\log^k n)$.

Remarks

- Exercise 4.6-3: $af(n/b) \le cf(n)$ for some constant $c<1$ and all sufficiently large n implies $f(n)=\Omega(n^{\log_b a+\varepsilon})$ for some constant $\varepsilon>0$.
- Question: Does $af(n/b) \ge cf(n)$ for some constant $c>1$ and all sufficiently large n imply $f(n)=O(n^{\log_b a-\varepsilon})$ for some constant $\varepsilon>0$?

Linear Recurrences

- General form: HARD to solve
  $T(n)=a_1T(n-1)+\ldots+a_kT(n-k)+f(n)$
- Simplest form: solvable by linear algebra
  $T(n)=a_1T(n-1)+\ldots+a_kT(n-k)+1$
- Example:
  $T(n)=T(n-1)+T(n-2)+1$ for $T(1)=T(2)=1$

Example

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} T(n-1) \\ T(n-2) \\ 1 \end{pmatrix} = \begin{pmatrix} T(n) \\ T(n-1) \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}^{n-2} \begin{pmatrix} T(2) \\ T(1) \\ 1 \end{pmatrix} = \begin{pmatrix} T(n) \\ T(n-1) \\ 1 \end{pmatrix}$$

$$T(n) = c_1\lambda_1^{n-2} + c_2\lambda_2^{n-2} + c_3\lambda_3^{n-2} = \Theta(\lambda_1^n)$$

$\boxed{\lambda_1, \lambda_2, \lambda_3 \text{ are eigenvalues where } |\lambda_1| \geq |\lambda_2| \geq |\lambda_3|}$

Homework

1. Solve $T(n) = T(\frac{2n}{3}) + T(\frac{n}{3}) + n$
2. Solve $T(n) = T(\frac{n}{3}) + T(\frac{2n}{5}) + n$
3. Solve $T(n) = \sqrt{n}T(\sqrt{n}) + n$
4. Solve $T(n) = 2T(n) + 1$
5. Solve $T(n) = T(n-1) + 2T(n-2) + 1$ for $T(0) = T(1) = 1$.
6. Solve $T(n) = T(\frac{n}{2}) + T(\frac{n}{4}) + 1$