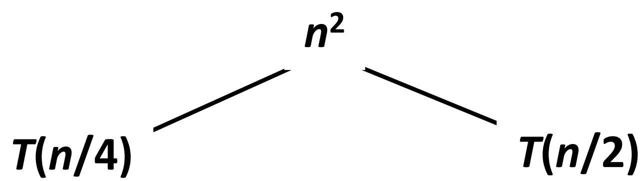
1. 采用Recursion-Tree方法求解递推式
   $T(n) = T(n/4) + T(n/2) + n^2$.

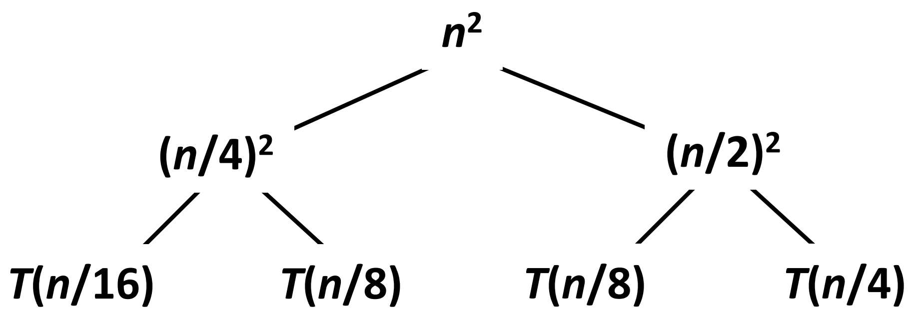
1. 采用Recursion-Tree方法求解递推式 $T(n) = T(n/4) + T(n/2) + n^2$.

$\boldsymbol{T(n)}$

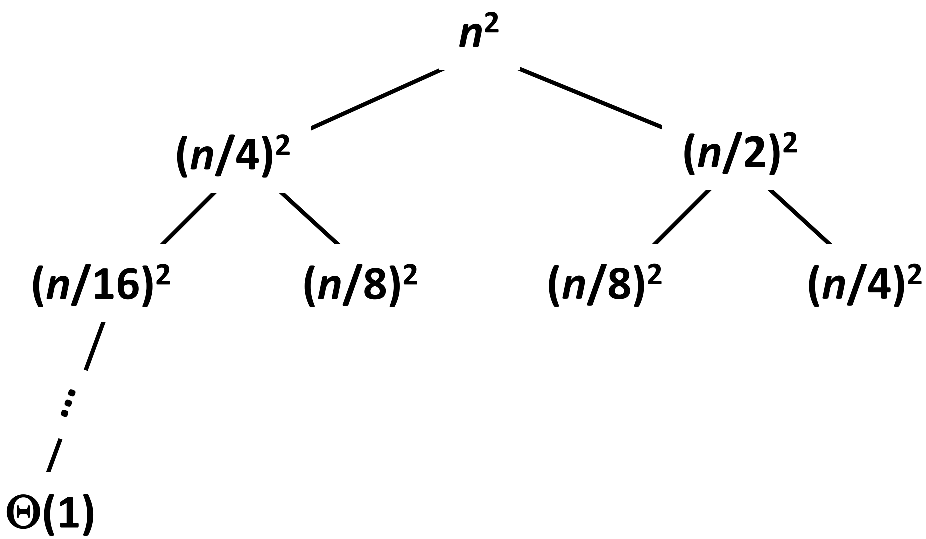
1. 采用Recursion-Tree方法求解递推式
   $T(n) = T(n/4) + T(n/2) + n^2$.



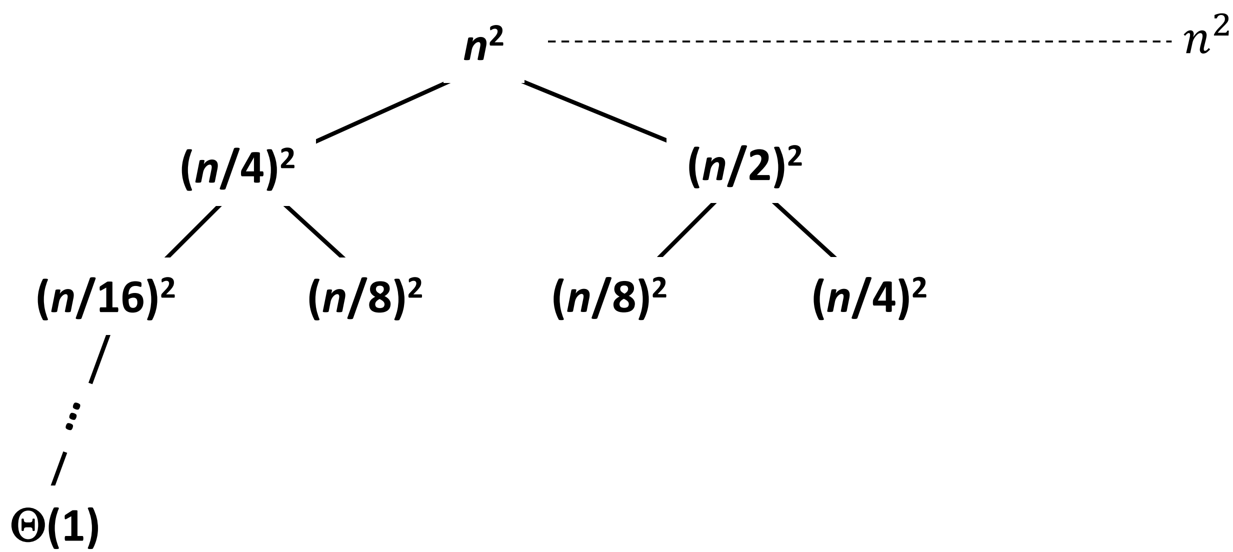
1. 采用Recursion-Tree方法求解递推式
   $T(n) = T(n/4) + T(n/2) + n^2$.



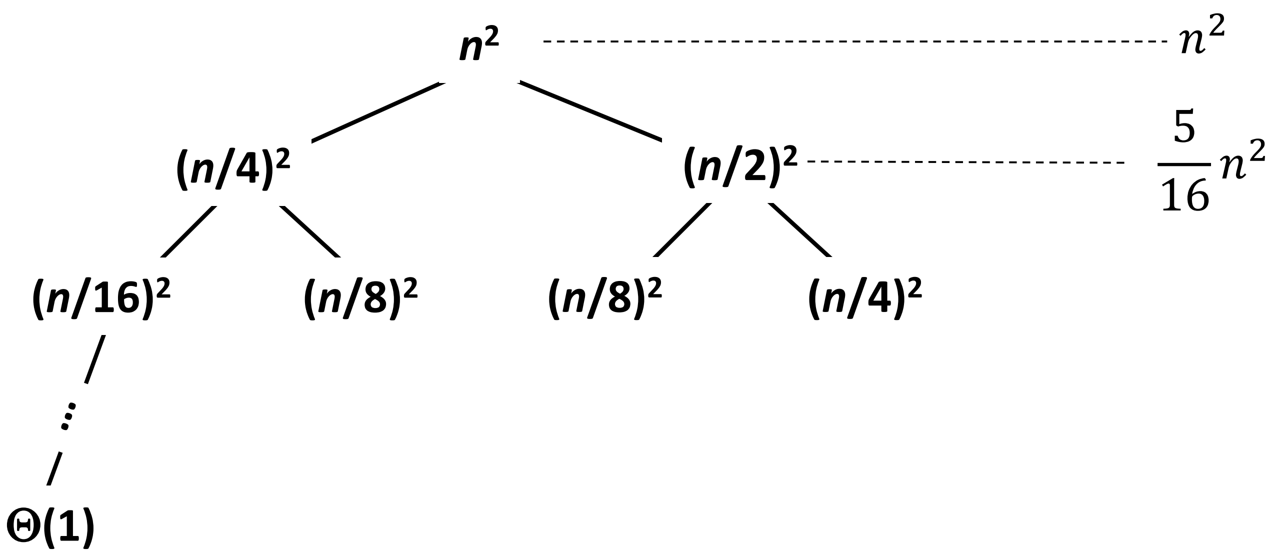
1. 采用Recursion-Tree方法求解递推式 $T(n) = T(n/4) + T(n/2) + n^2$.



1. 采用Recursion-Tree方法求解递推式
$T(n) = T(n/4) + T(n/2) + n^2$.

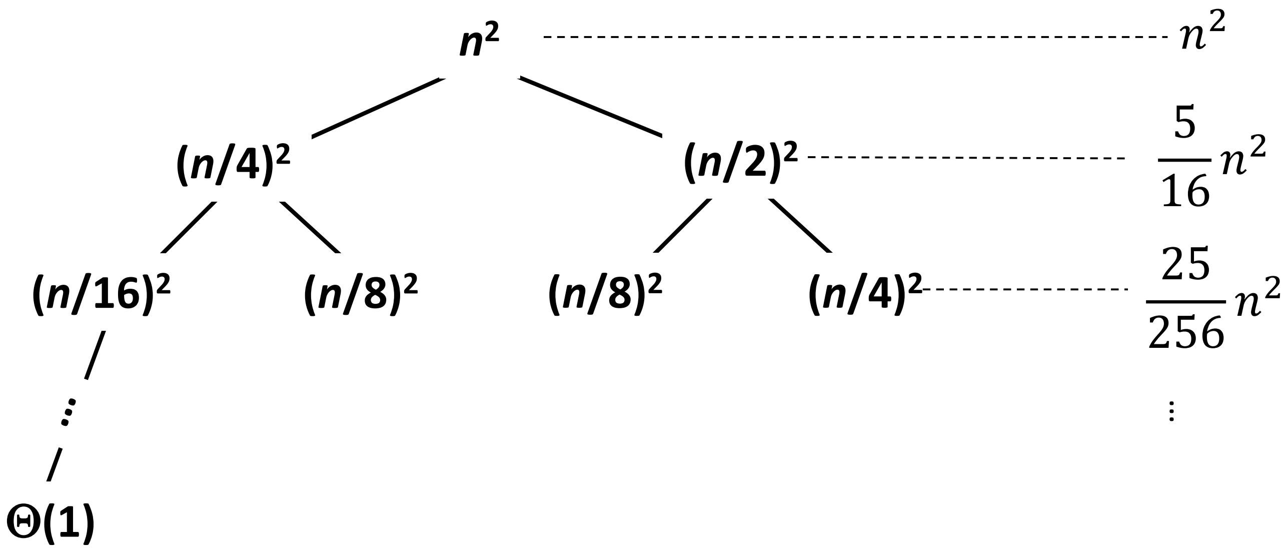


1. 采用Recursion-Tree方法求解递推式 $T(n) = T(n/4) + T(n/2) + n^2$.

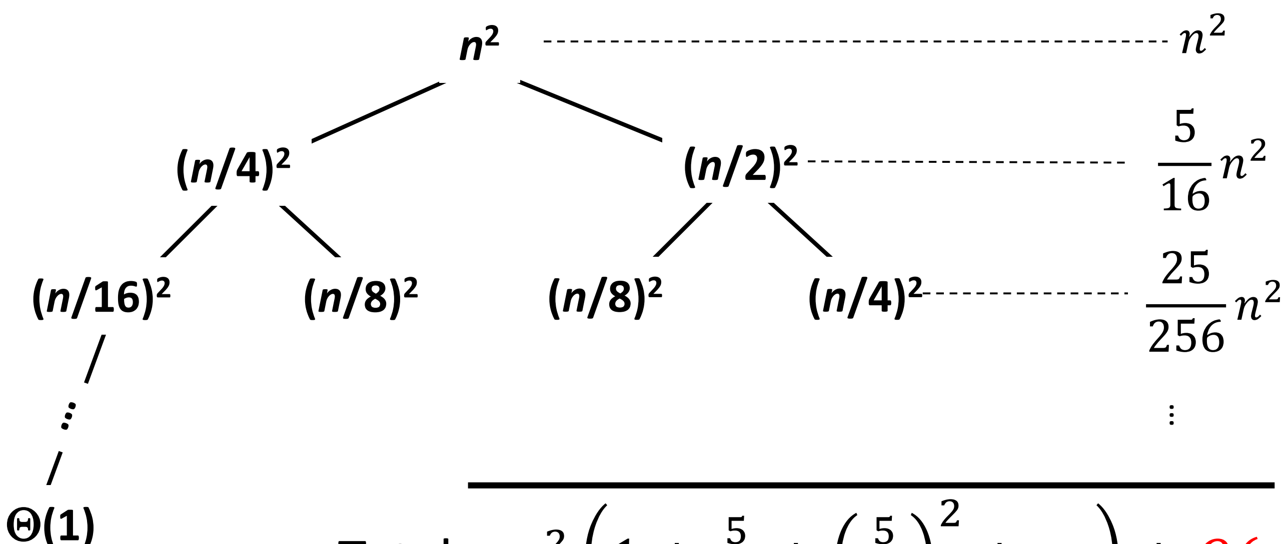


1. 采用Recursion-Tree方法求解递推式

$T(n) = T(n/4) + T(n/2) + n^2$.



# 1. 采用Recursion-Tree方法求解递推式 $T(n) = T(n/4) + T(n/2) + n^2$.




Total = $n^2\left(1 + \frac{5}{16} + \left(\frac{5}{16}\right)^2 + \cdots\right) + O(n)$

= $\Theta(n^2)$ *geometric series*

# 2. 采用Master Theorem求解下列递推式.

*a) T(n) = 4T(n/2) + n*

*b) T(n) = 4T(n/2) + n²*

*c) T(n) = 4T(n/2) + n³*

*d) T(n)=3T(n/4)+nlogn*

Master Theorem. Suppose that $T(n)$ is a function on the non-negative integers that satisfies the recurrence:

$$T(n) = aT(n/b) + f(n)$$

with $T(0) = 0\ and\ T(1) = \Theta(1)$, where $n/b$ means either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$ .

Case 1. If $f(n) = O(n^k)$ for some constant $k < \log_b a$, then $T(n) = \Theta\left(n^{\log_b a}\right)$.

Case 2. If $f(n) = \Theta(n^k log^p n)$ for $p \geq 0$ and $k = \log_b a$, then $T(n) = \Theta\left(n^k log^{p+1} n\right)$.

Case 3. If $f(n) = \Omega(n^k)$ for some constant $k > \log_b a$, and if $af(n/b) \leq cf(n)$ for some constant $c < 1$ and all sufficiently large $n$, then $T(n) = \Theta\left(f(n)\right)$.

# 2. 采用Master Theorem求解下列递推式.

*a)* $T(n) = 4T(n/2) + n$

$a = 4$, $b = 2$,

$n^{\log_b a} = n^2$; *and*

$f(n) = n = O(n^1)$.

*Case 1:* $k < \log_b a$,

*then* $T(n) = \Theta(n^2)$.

Master Theorem. Suppose that $T(n)$ is a function on the non-negative integers that satisfies the recurrence:

$$T(n) = aT(n/b) + f(n)$$

with $T(0) = 0 \text{ and } T(1) = \Theta(1)$, where $n/b$ means either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$ .

Case 1. If $f(n) = O(n^k)$ for some constant $k < \log_b a$, then $T(n) = \Theta\left(n^{\log_b a}\right)$.

Case 2. If $f(n) = \Theta(n^k log^p n)$ for $p \geq 0$ and $k = \log_b a$, then $T(n) = \Theta\left(n^k log^{p+1} n\right)$.

Case 3. If $f(n) = \Omega(n^k)$ for some constant $k > \log_b a$, and if $af(n/b) \leq cf(n)$ for some constant $c < 1$ and all sufficiently large $n$, then $T(n) = \Theta\left(f(n)\right)$.

# 2. 采用Master Theorem求解下列递推式.

*b)* $T(n) = 4T(n/2) + n^2$

$a = 4, b = 2,$
$n^{\log_b a} = n^2$; *and*
$f(n) = n^2$.
*Case 2:* $f(n) = (n^2 \log^0 n)$,
*and* $k = 2$,
*then* $T(n) = \Theta(n^2 \log n)$.

Master Theorem. Suppose that $T(n)$ is a function on the non-negative integers that satisfies the recurrence:

$$T(n) = aT(n/b) + f(n)$$

with $T(0) = 0 \; and \; T(1) = \Theta(1)$, where $n/b$ means either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$ .

Case 1. If $f(n) = O(n^k)$ for some constant $k < \log_b a$, then $T(n) = \Theta\left(n^{\log_b a}\right)$.

Case 2. If $f(n) = \Theta(n^k log^p n)$ for $p \geq 0$ and $k = \log_b a$, then $T(n) = \Theta\left(n^k log^{p+1} n\right)$.

Case 3. If $f(n) = \Omega(n^k)$ for some constant $k > \log_b a$, and if $af(n/b) \leq cf(n)$ for some constant $c < 1$ and all sufficiently large $n$, then $T(n) = \Theta\left(f(n)\right)$.

# 2. 采用Master Theorem求解下列递推式.

*c)* $T(n) = 4T(n/2) + n^3$

$a = 4$, $b = 2$,
$n^{\log_b a} = n^2$; *and*
$f(n) = n^3$.
*Case 3:* $f(n) = \Omega(n^3)$,
*and* $4(n/2)^3 \leq cn^3$ *(reg. cond.) for* $c = ½$,
*then* $T(n) = \Theta(n^3)$.

Master Theorem. Suppose that $T(n)$ is a function on the non-negative integers that satisfies the recurrence:

$$T(n) = aT(n/b) + f(n)$$

with $T(0) = 0 \ and \ T(1) = \Theta(1)$, where $n/b$ means either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$ .

Case 1. If $f(n) = O(n^k)$ for some constant $k < \log_b a$, then $T(n) = \Theta\left(n^{\log_b a}\right)$.

Case 2. If $f(n) = \Theta(n^k log^p n)$ for $p \geq 0$ and $k = \log_b a$, then $T(n) = \Theta\left(n^k log^{p+1} n\right)$.

Case 3. If $f(n) = \Omega(n^k)$ for some constant $k > \log_b a$, and if $af(n/b) \leq cf(n)$ for some constant $c < 1$ and all sufficiently large $n$, then $T(n) = \Theta\left(f(n)\right)$.

# 2. 采用Master Theorem求解下列递推式.

d) *T(n)=3T(n/4)+nlogn*

*a = 3, b = 4,*
$n^{\log_b a} = n^{0.793}$*; and*
*f (n) = nlogn.*
*Case 3:* $f(n) = \Omega(n^1)$*,*
*and* $3(n/4)\log(n/4) \leq cn\log n$ *(reg. cond.) for*
*c = 3/4,*
*then* $T(n) = \Theta(n\log n)$*.*

Master Theorem. Suppose that $T(n)$ is a function on the non-negative integers that satisfies the recurrence:

$$T(n) = aT(n/b) + f(n)$$

with $T(0) = 0$ *and* $T(1) = \Theta(1)$, where $n/b$ means either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$ .

Case 1. If $f(n) = O(n^k)$ for some constant $k < \log_b a$, then $T(n) = \Theta\left(n^{\log_b a}\right)$.

Case 2. If $f(n) = \Theta(n^k log^p n)$ for $p \geq 0$ and $k = \log_b a$, then $T(n) = \Theta\left(n^k log^{p+1} n\right)$.

Case 3. If $f(n) = \Omega(n^k)$ for some constant $k > \log_b a$, and if $af(n/b) \leq cf(n)$ for some constant $c < 1$ and all sufficiently large $n$, then $T(n) = \Theta\left(f(n)\right)$.