Linearly decreasing time functions:

$T(n) = T(n-1) + 1 \quad \sim O(n)$
$T(n) = T(n-1) + n \quad \sim O(n^2)$

$T(n) = T(n-1) + \log n \quad \sim O(n\log n)$

$T(n-1) = T(n-2) + \log (n-1)$
$T(n-2) = T(n-3) + \log (n-2)$

$T(n) = T(n-2) + \log (n-1) + \log n$
$\quad = T(n-3) + \log (n-2) + \log (n-1) + \log n$
$\quad \ldots \quad n - k = 0$
$\quad = T(n-n) + \log 1 + \log 2 + \log 3 + \ldots + \log n$
$\quad = T(0) + \log n!$
$\quad = 1 + \log n!$
$\quad \sim$ **$O(n\log n)$**

…

Master Theorem for Linear Decreasing Functions:

$T(n) = \mathbf{a}T(n - \mathbf{b}) + \mathbf{f(n)}$
a > 0, **b** > 0, and **$f(n) = O(n^k)$ where $k >= 0$**

Case1: if $a = 1$, $O(f(n) * \mathbf{n})$
Case2: if $a > 1$, $O(f(n)*\mathbf{a^{n/b}})$
Case3: if $a < 1$, $O(f(n))$ ← *too small to consider*

Case1:

$T(n) = T(n-1) + n\log n \quad \sim O(n^2\log n)$
$T(n) = T(n-1) + n^2 \quad \sim O(n^3)$
$T(n) = T(n-1) + n^3 \quad \sim O(n^4)$
$T(n) = T(n-2) + 1 \quad \sim O(n)$
$T(n) = T(n-1000) + n \quad \sim O(n^2)$

Case2:

$T(n) = 2T(n-1) + 1 \quad \sim O(2^n)$
$T(n) = 3T(n-1) + 1 \quad \sim O(3^n)$
$T(n) = 2T(n-1) + n \quad \sim O(n2^n)$
$T(n) = 2T(n-1) + n^3 \quad \sim O(n^3 2^n)$
$T(n) = 2T(n-2) + 1 \quad \sim O(2^{n/2})$

Case3:

$T(n) = 1/2T(n-1) + 1 \quad \sim O(1)$
$T(n) = 1/2T(n-1) + n \quad \sim O(n)$

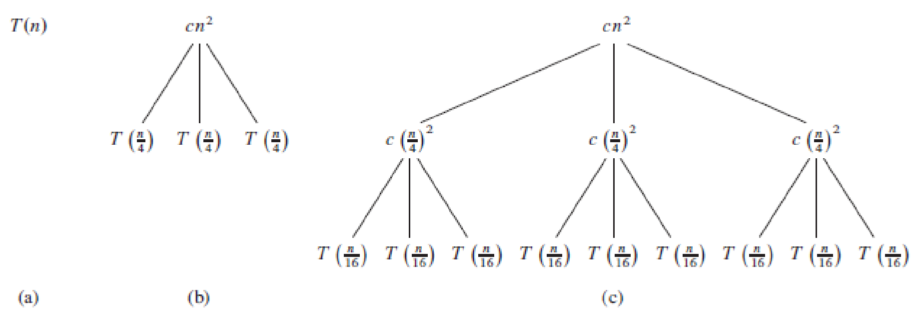
Dividing time functions (finding a dominating part):

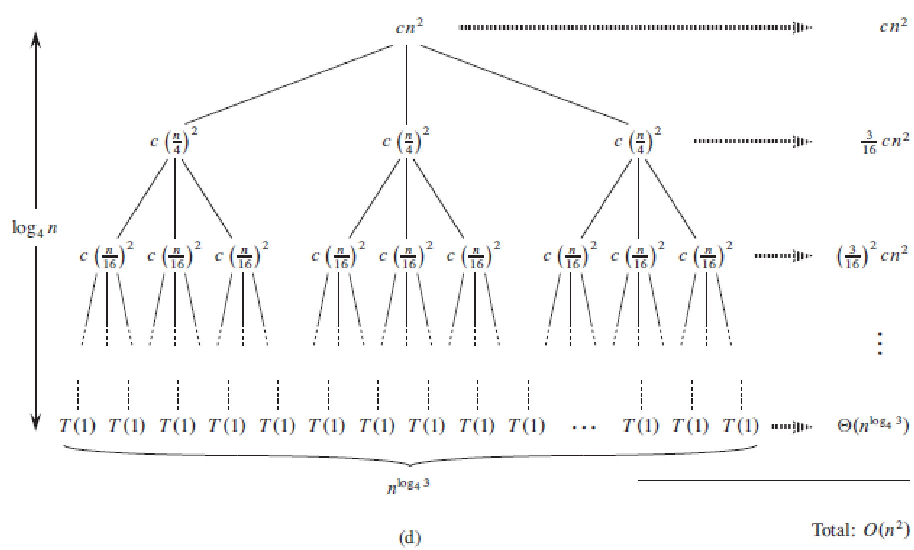
Left (T) Dominate:
T(n) = T(n/2) + 1 ~ O(lgn)
T(n) = 2T(n/2) + 1 ~ O(n)
T(n) = 2T(n/2) + $\sqrt{n}$ ~ O(n)
T(n) = 4T(n/4) + 1 ~ O(n)

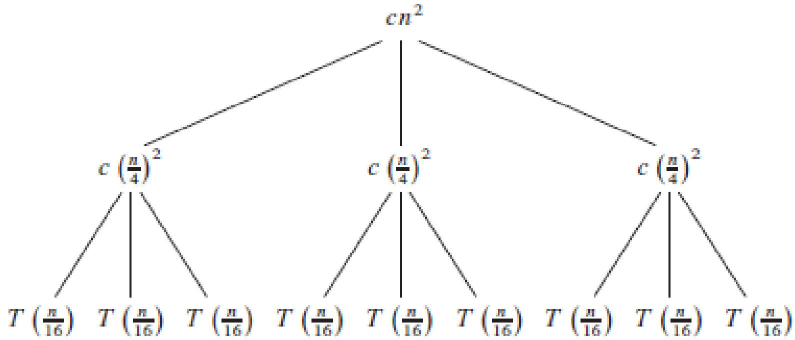
T(n) = 16T(n/4) + n $\Rightarrow$T(n) = O(n^2)
T(n) = 4T(n/2) + 1 ~ O(n^2)
T(n) = 8T(n/2) + 1 ~ O(n^3)

Right Dominate:
T(n) = T(n/2) + n ~ O(n)
T(n) = T(n/2) + n^2 $\Rightarrow$T(n) = O(n^2)
T(n) = 2T(n/2) + n^2 ~ O(n^2)
T(n) = 2T(n/2) + n^3 ~ O(n^3)
T(n) = 2T(n/2) + nlgn ~ O(nlgn) ? → **O(nlog^{2}n)**
T(n) = 2T(n/4) + $n^{0.51}$ $\Rightarrow$T(n) = O($n^{0.51}$)

T(n) = 3T(n/4) + n^2 ~ O(n^2) ← figure 4.5 in the text book
3 is a number of branches; 4 defines the tree depth; n^2 is each iteration.







$T(n) = 3T(n/4) + n\lg n \sim O(n\lg n)$

(When there is a complex log, we need a formal rule)

Equivalent:
$T(n) = 2T(n/2) + n \sim O(n\lg n)$
$T(n) = 4T(n/2) + n^2 \quad \Rightarrow T(n) = O(n^2 \lg n)$

Master Theorem for Dividing Functions:

$T(n) = \mathbf{a}T(n/\mathbf{b}) + \mathbf{f(n)}$

a >=1, **b** > 1, and $\mathbf{f(n) = O(n^k \log^p n)}$
Check the condition of $\mathbf{\log_b a}$ and $\boldsymbol{k}$

Intuitively, **a** is a number of branches; **b** defines the tree depth; **f(n)** is an iteration of each node

case 1:
if $\log_b a > k$, $O(n^{\log_b a})$ ← left (T) dominate

case 2:
if $\log_b a = k$, further condition of p: ← Equivalent
if $p > -1$, $O(n^k \log^{p+1} n)$ ← in general f(n) * logn
if $p = -1$, $O(n^k \log\log n)$
if $p < -1$, $O(n^k)$ ← ignore the log which is too small

case 3:
if $\log_b a < k$, further condition of p: ← right dominate
if $p >= 0$, $O(n^k \log^p n)$ ← use f(n) as it is
if $p < 0$, $O(n^k)$ ← ignore the log which is too small

case 1: $T(n) = 2T(n/2) + 1 \sim O(n)$
case 1: $T(n) = 4T(n/2) + n \sim O(n^2)$
case 1: $T(n) = 8T(n/2) + n \sim O(n^3)$
case 1: $T(n) = 8T(n/2) + n^2 \sim O(n^3)$
case 1: $T(n) = 8T(n/2) + n\lg n \sim O(n^3)$
case 1: $T(n) = 9T(n/3) + 1 \sim O(n^2)$
case 1: $T(n) = 9T(n/3) + n \sim O(n^2)$

case 3: $T(n) = 2T(n/2) + n^2 \sim O(n^2)$, $\log_2 2 = 1, k = 2, p= 0$
case 3: $T(n) = 2T(n/2) + n^2 \log^2 n \sim O(n^2 \log^2 n)$, $\log_2 2 = 1, k = 2, p= 2$

case 3-2: T(n) = 2T(n/2) + $n^2 / \log^2 n$ ~ O($\mathbf{n^2}$), ***$\log_2 2$ = 1, k = 2, p= -2***

case 2-1: T(n) = 9T(n/3) + n^2 ~ O($n^2 \log n$), ***$\log_3 9$ = 2, k = 2, p= 0***
case 2-1: T(n) = 4T(n/2) + n^2 ~ O($n^2 \log n$), ***$\log_2 4$ = 2, k = 2, p= 0***
case 2-1: T(n) = 2T(n/2) + n ~ O(n logn), ***$\log_2 2$ = 1, k = 1, p= 0***
case 2-1: T(n) = 4T(n/2) + $n^2 \log^2 n$ ~ O($n^2 \log^3 n$), ***$\log_2 4$ = 2, k = 2, p= 2***
case 2-2: T(n) = 2T(n/2) + n/logn ~ O(nloglogn), ***$\log_2 2$ = 1, k = 1, p= -1***
case 2-3: T(n) = 4T(n/2) + $n^2 / \log^2 n$ ~ O(n^2), ***$\log_2 4$ = 2, k = 2, p= -2***

Exercises:

T(n) = 2T(n/2) + n^3 ~ O(n^3) case 3-1
T(n) = T(9n/10) + n ~ O(n) a = 1, b = 10/9, k =1, p=0, $\mathbf{n}^{\log_{10/9} 1}$, $(10/9)^0 = 1$, $n^0 = 1$, case 3-1
T(n) = 16T(n/4) + n^2 ~ O(n^2 logn), case 2
T(n) = 7T(n/3) + n^2 ~ O(n^2), $\log_3 7 < \log_3 9 = 2$, case 3-1
T(n) = 7T(n/2) + n^2 ~ O($n^{\log 7}$), $\log_2 7 > \log_2 4 = 2$, case 1
T(n) = 2T(n/4) + $\sqrt{n}$ ~ O($\sqrt{n}$lgn), $\mathbf{n}^{\log_4 2} = \mathbf{n}^{1/2} = \sqrt{n}$, case 2-1
T(n) = 4T(n/2) + $n^2 \sqrt{n}$ ~ O($n^2 \sqrt{n}$), $\mathbf{n}^{\log_2 4} = \mathbf{n}^2 < \mathbf{n}^{5/2}$, k= 5/2, p=0, case 3-1
T(n) = 3T(n/2) + nlgn ~ O($n^{\log 3}$), $\log_2 3 > \log_2 2$, k =1, p =1, case 1