Design and Analysis of Algorithms

Tutorial 1: Asymptotic Notations and Recurrences

许可 kexu@nlsde.buaa.edu.cn

童咏昕 yxtong@buaa.edu.cn

北京航空航天大学 计算机学院

复习：渐近记号

近似上界：big-Oh

- f(n)=O(g(n))：存在常数c > 0和n_0，使得对于$n \ge n_0$有 $f(n) \le c \cdot g(n)$

近似下界：big-Omega

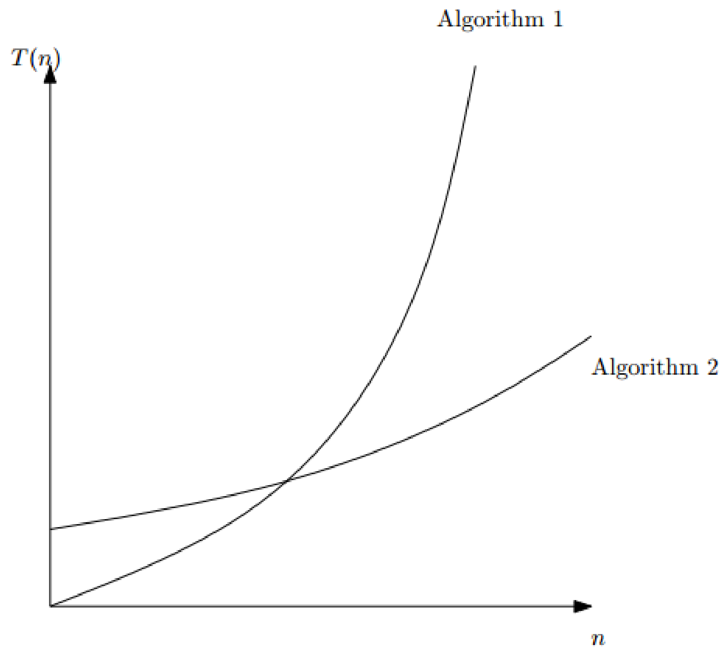
- f (n)=Ω(g(n))：存在常数c > 0和n_0，使得对于$n \ge n_0$ 有$f(n) \ge c \cdot g(n)$.

近似紧界：big-Theta

- f(n)=Θ(g(n))：f(n)=O(g(n)) 且f(n)=Ω(g(n)).

复习：时间复杂度

例:



显然Algorithm 2更好

- Algorithm 1 的近似上界为$O(n^3)$
- Algorithm 2 的近似上界为$O(n^2)$
- 由于n^3 增长速度更快，我们认为当n增加时Algorithm 1 比Algorithm 2花费更多时间。

复习：指数函数基础

- 对于实数a ≠ 0，m 以及n，有如下性质：

$$\mathrm{a}^0 = 1$$

$$a^1 = a$$

$$a^{-1} = \frac{1}{a}$$

$$(a^m)^n = (a^n)^m = a^{mn}$$

$$a^m a^n = a^{m+n}$$

$$a^{\frac{1}{n}} = \sqrt[n]{a}$$

复习：对数函数基础

- 对于实数a ＞ 0， b ＞ 0， c ＞ 0 和n：

$$a = b^{\log_b a}$$

$$\log_c(ab) = \log_c a + \log_c b$$

$$\log_b a^n = nlog_b a$$

$$\log_b a = \frac{\log_c a}{\log_c b}$$

$$\log_b(\frac{1}{a}) = -\log_b a$$

$$\log_b a = \frac{1}{\log_a b}$$

$$a^{\log_b n} = n^{\log_b a}$$

问题1

- 判断并证明下面给出的每一个表达式对(A, B)是否满足A=O(B)，A=Ω(B)或A=Θ(B)．注意：一个表达式对可能满足多个关系，请列出所有可能的结果。

(a) $A = n^3 + nlogn; B = n^3 + n^2logn$

(b) $A = log\sqrt{n}; B = \sqrt{logn}$

(c) $A = nlog_3n; B = nlog_4n$

(d) $A = 2^n; B = 2^{\frac{n}{2}}$

(e) $A = \log(2^n); B = \log(3^n)$

问题1–提示

	A	关系:	B
(a)	$n^3+nlogn$	Ω,Θ,O	n^3+n^2logn
(b)	$log\sqrt{n}$	Ω	$\sqrt{logn}$
(c)	$nlog_3n$	Ω,Θ,O	$nlog_4n$
(d)	2^n	Ω	$2^{\frac{n}{2}}$
(e)	$\log(2^n)$	Ω,Θ,O	$\log(3^n)$

问题1-提示

证明:

（a）A与B两个式子都属于 $\Theta(n^3)$, 低次项可省略. 同时注意到，若$A(n) = \Theta(B(n))$, 则有$A(n) = O(B(n))$和$A(n) = \Omega(B(n))$成立.

（b）A化简为$(1/2)\log n$，而B为$\sqrt{logn}$. 令$m = \log n$，则$\frac{\mathrm{A}}{\mathrm{B}} = \frac{m}{2\sqrt{m}} = \frac{\sqrt{m}}{2}$，该值当n趋向于无穷大时亦趋向于无穷大。因此，$A(n) = \Omega(B(n))$.

（c）对数的底数转换只会导致常数因子，因此$A = \Theta(B)$.

（d）$\frac{\mathrm{A}}{\mathrm{B}} = \frac{2^n}{2^{\frac{n}{2}}} = (2)^{\frac{n}{2}}$, 该值当n趋向于无穷大时趋向于无穷大。

（e）经过化简，$A = n\log 2$，$B = n\log 3$，二者都属于$\Theta(n)$.

问题2

- 假设 $T_1(n) = O(f(n))$, $T_2(n) = O(f(n))$. 下列公式是否正确? 请证明你的答案。

（a） $T_1(n) + T_2(n) = O\big(f(n)\big)$

（b） $\frac{T_1(n)}{T_2(n)} = O(1)$

（c） $T_1(n) = O(T_2(n))$

问题2-提示

（a）正确．由于 $T_1(n) = O(f(n))$且$T_2(n) = O(f(n))$, 由定义知存在常数$c_1, c_2 > 0$ 和正整数n_1, n_2 , 使得对任意$n \geq n_1$, 有$T_1(n) \leq c_1 f(n)$;对任意$n \geq n_2$有$T_2(n) \leq c_2 f(n)$. 进而对任意$n \geq \max(n_1, n_2), T_1(n) + T_2(n) \leq (c_1 + c_2) f(n)$．因此$T_1(n) + T_2(n) = O(f(n))$.

（b）错误．反例如下：令$T_1(n) = n^2$，$T_2(n) = n$，$f(n) = n^2$．则有 $T_1(n) = O(f(n))$和$T_2(n) = O(f(n))$, 然而 $\frac{\mathrm{T}_1(n)}{T_2(n)} = n \neq O(1)$.

（c）错误．反例见(b)。

问题3

- 使用递归树(recursion tree)的方法计算下列递归函数T(n)的最紧渐近上界:

(a)

$$T(1) = 1$$
$$T(n) = T\left(\frac{n}{2}\right) + n \qquad if\ n > 1$$

(b)

$$T(1) = T(2) = 1$$
$$T(n) = T(n-2) + 1 \quad if\ n > 2$$

(c)

$$T(1) = 1$$
$$T(n) = T\left(\frac{n}{3}\right) + n \qquad if\ n > 1$$

问题3

- 使用递归树(recursion tree)的方法计算下列递归函数T(n)的最紧渐近上界:

(d)

$$T(1) = 1$$
$$T(n) = 4T\left(\frac{n}{2}\right) + n \qquad if\ n > 1$$

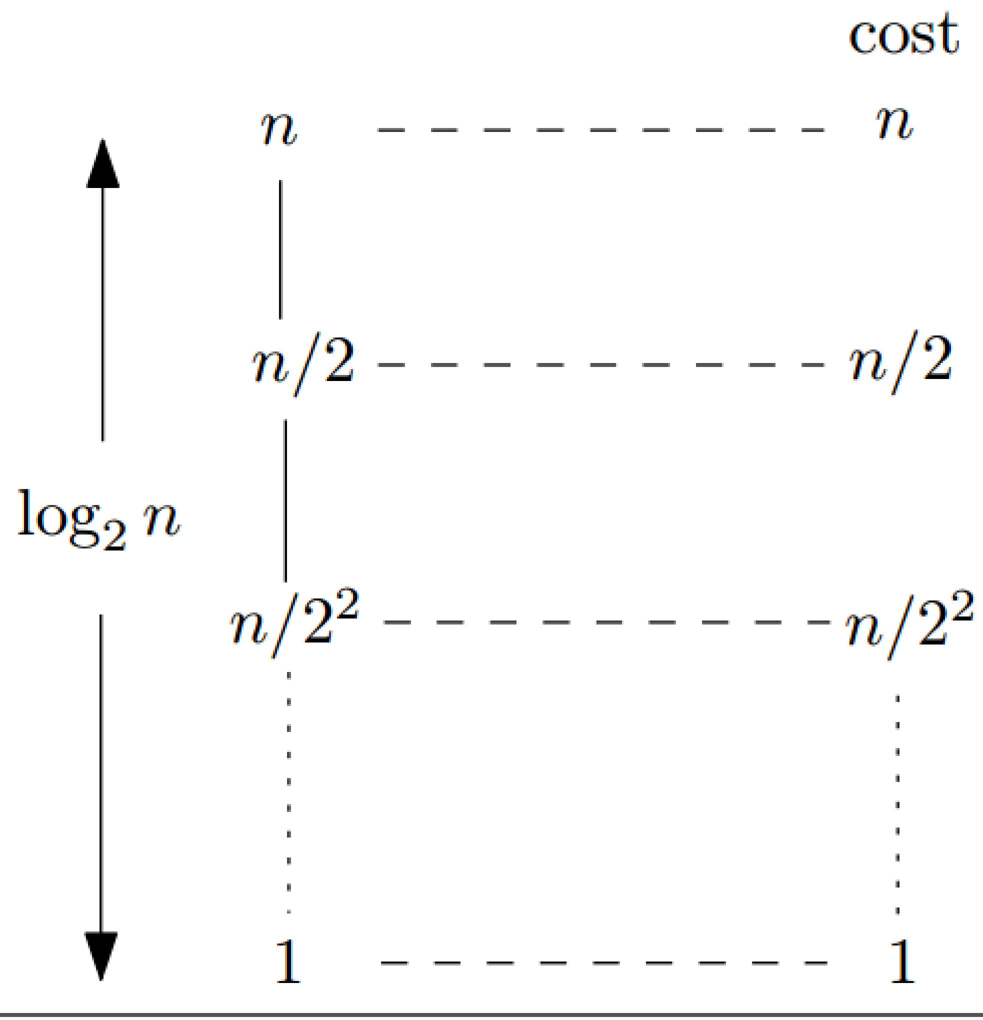
(e)

$$T(1) = 1$$
$$T(n) = 3T\left(\frac{n}{2}\right) + n^2 \qquad if\ n > 1$$

(f)

$$T(1) = 0, T(2) = 1$$
$$T(n) = T\left(\frac{n}{2}\right) + \log_2 n \qquad if\ n > 2$$

问题3(a)-提示



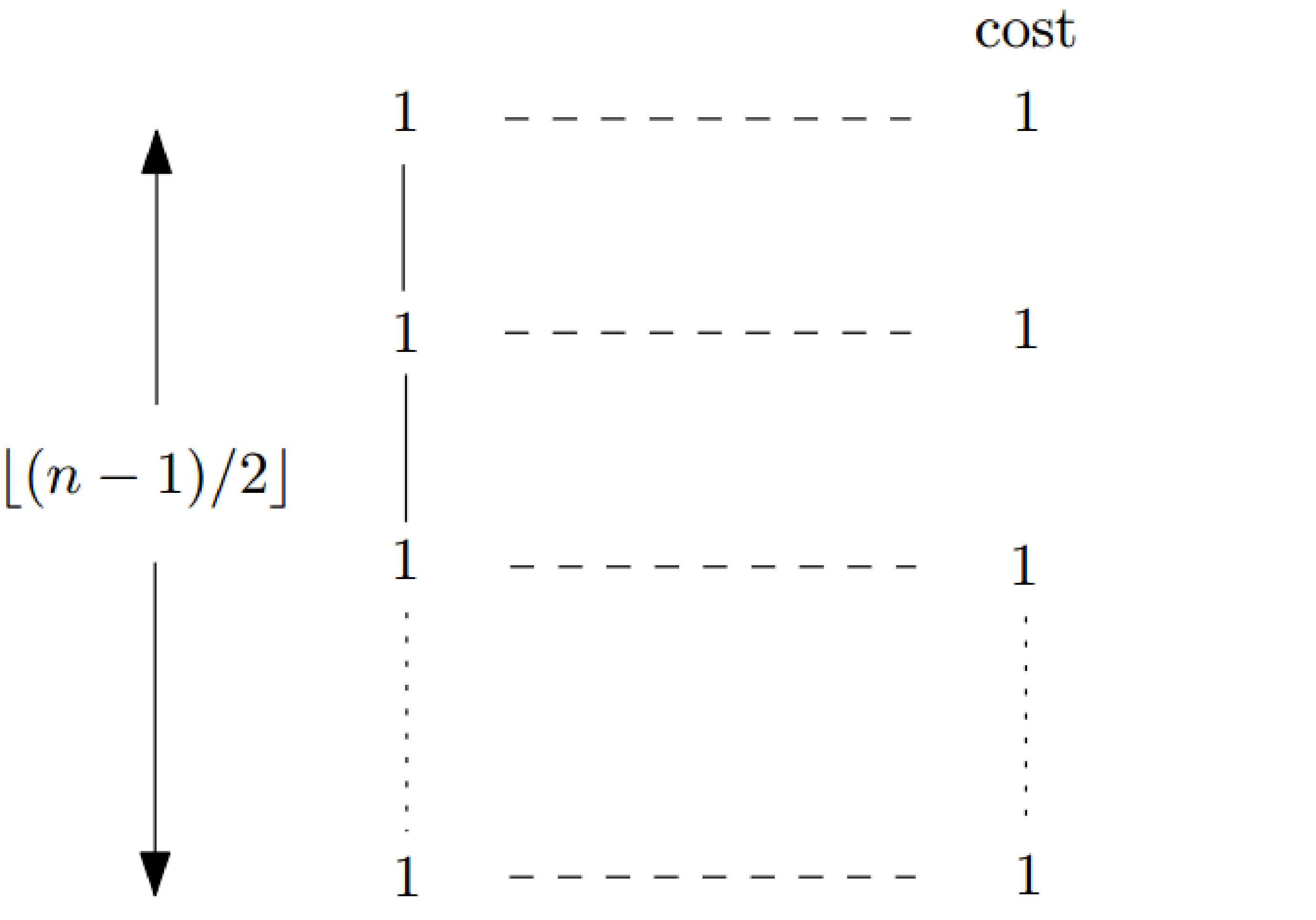
$$\text{total} = n + n/2 + n/2^2 \cdots + 1$$
$$= O(n)$$

问题3 (a) –提示

Set $h = \log_2 n$

$$
\begin{aligned}
T(n) &= n + T(n/2) \\
&= n + n/2 + T(n/2^2) \\
&= n + n/2 + n/2^2 + T(n/2^3) \\
&\dots \\
&= n + n/2 + n/2^2 + \cdots + n/2^{h-2} + n/2^{h-1} + T(n/2^h) \\
&= n(1 + 1/2 + 1/2^2 + \cdots + 1/2^{h-2} + 1/2^{h-1}) + T(n/2^h) \\
&\le n(1 + 1/2 + 1/2^2 + \cdots + 1/2^{h-1} + \dots) + T(n/2^h) \\
&= 2 \cdot n + T(1) \\
T(n) &= O(n)
\end{aligned}
$$

问题3(b)-提示

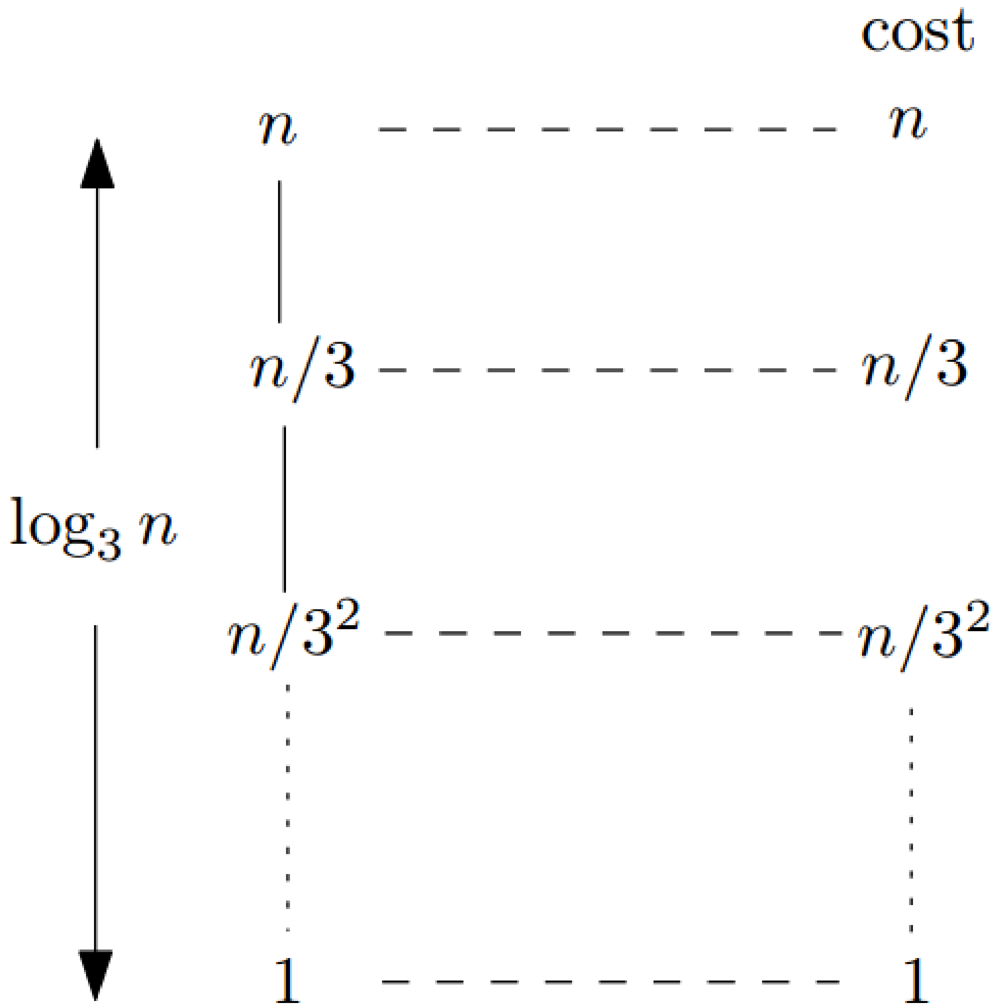


$$\text{total} = 1 + \lfloor (n-1)/2 \rfloor$$
$$= O(n)$$

问题3(b)−提示

$$
\begin{aligned}
T(n) &= T(n-2)+1 \\
&= T(n-2\cdot 2)+2 \\
&= T(n-3\cdot 2)+3 \\
&\cdots \\
&= T(n-\lfloor (n-1)/2\rfloor \cdot 2)+\lfloor (n-1)/2\rfloor \\
T(n) &= 1+\lfloor (n-1)/2\rfloor = \lceil (n/2)\rceil = O(n)
\end{aligned}
$$

问题3(c)-提示



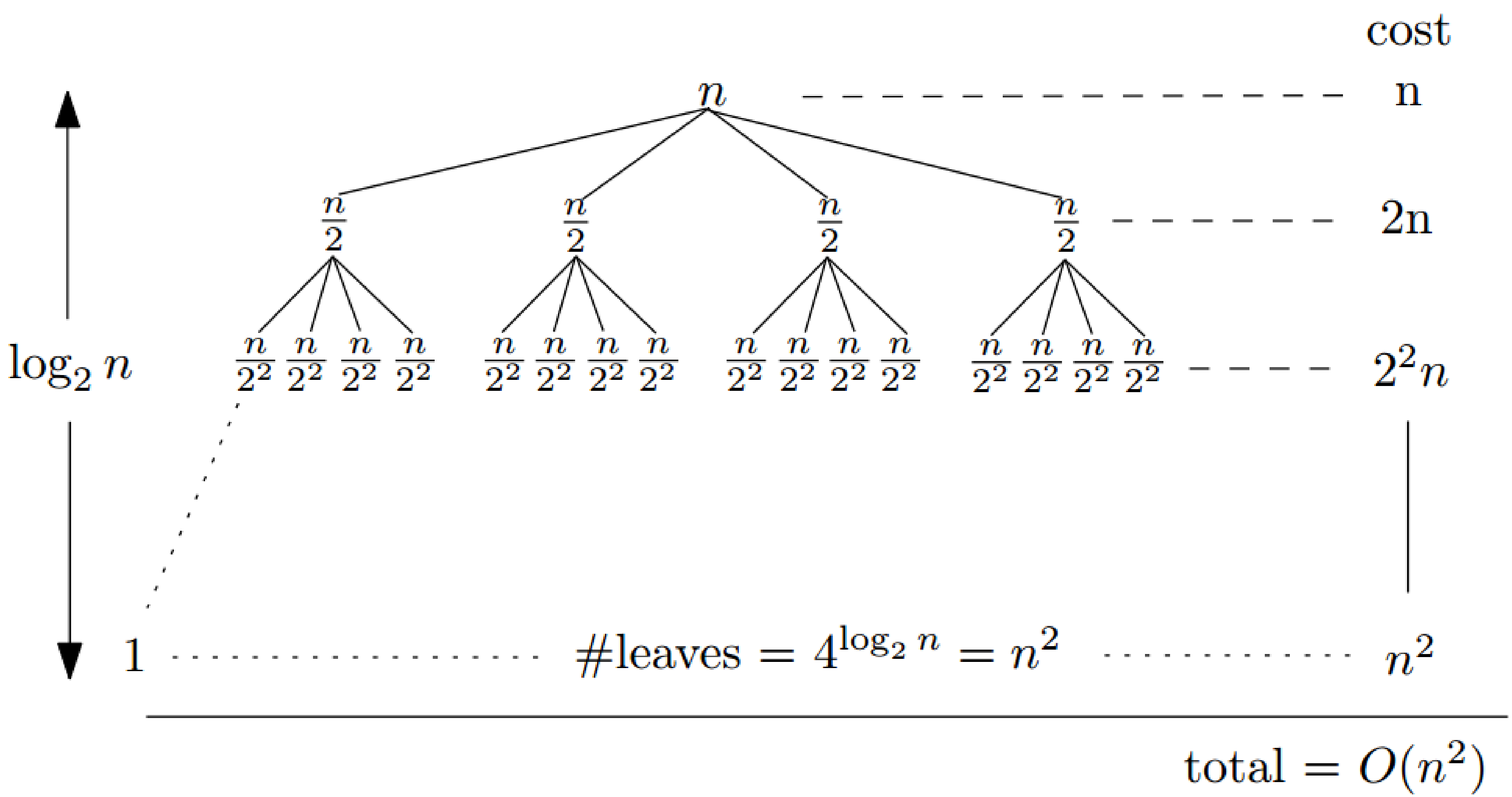
$$\text{total} = n + n/3 + n/3^2 \cdots + 1$$
$$= O(n)$$

问题3 (c) –提示

Set $h = \log_3 n$

$$
\begin{aligned}
T(n) &= n + T(n/3) \\
&= n + n/3 + T(n/3^2) \\
&= n + n/3 + n/3^2 + T(n/3^3) \\
&\dots \\
&= n + n/3 + n/3^2 + \cdots + n/3^{h-2} + n/3^{h-1} + T(n/3^h) \\
&= n(1 + 1/3 + 1/3^2 + \cdots + 1/3^{h-2} + 1/3^{h-1}) + T(n/3^h) \\
&\leq n(1 + 1/3 + 1/3^2 + \cdots + 1/3^{h-1} + \dots) + T(n/3^h) \\
&= 3n/2 + T(1) \\
T(n) &= O(n)
\end{aligned}
$$

问题3(d)-提示

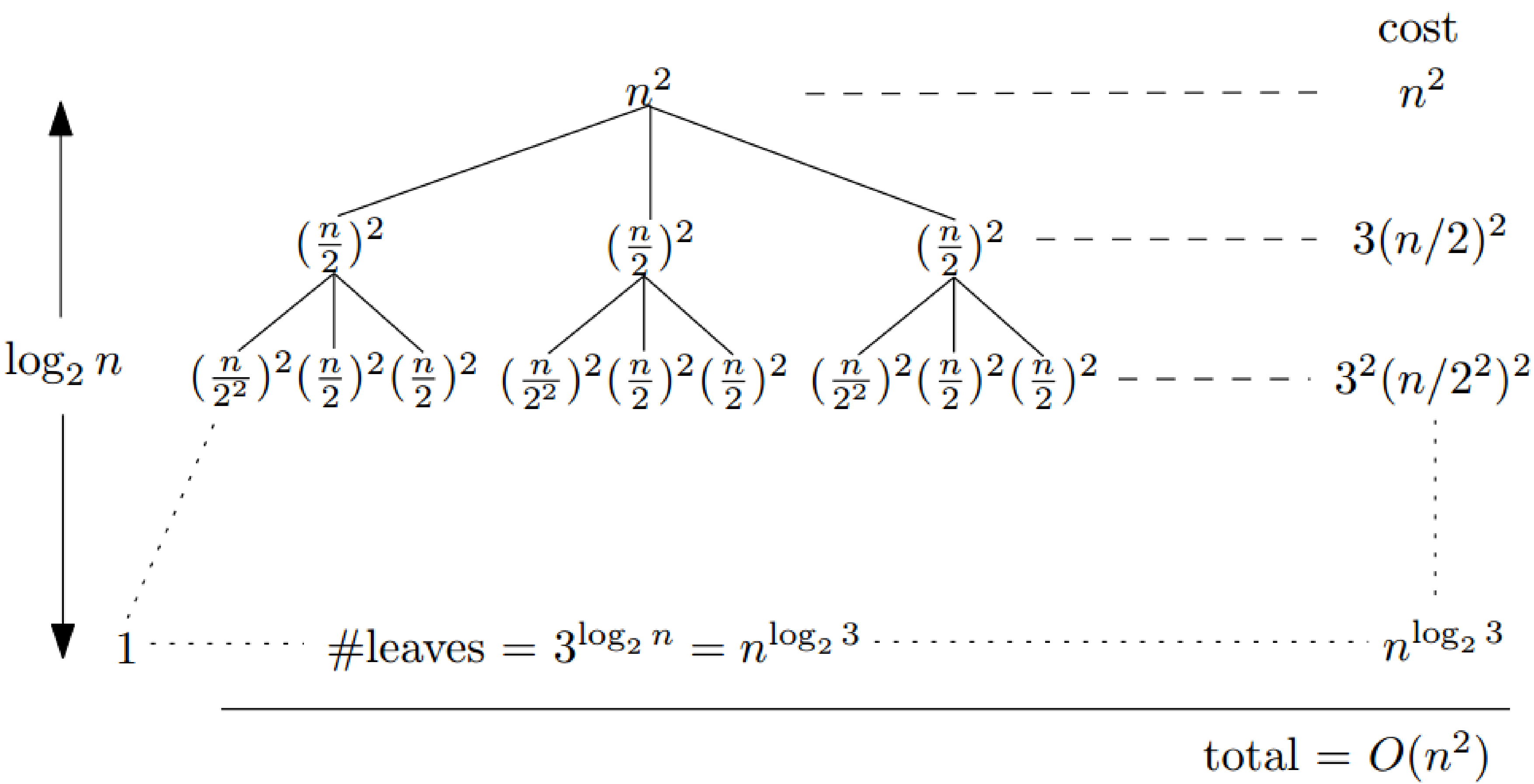

cost
n
n
$\frac{n}{2}$ $\frac{n}{2}$ $\frac{n}{2}$ $\frac{n}{2}$
2n
$\log_2 n$
$\frac{n}{2^2}$ $\frac{n}{2^2}$ $\frac{n}{2^2}$ $\frac{n}{2^2}$ $\frac{n}{2^2}$ $\frac{n}{2^2}$ $\frac{n}{2^2}$ $\frac{n}{2^2}$ $\frac{n}{2^2}$ $\frac{n}{2^2}$ $\frac{n}{2^2}$ $\frac{n}{2^2}$ $\frac{n}{2^2}$ $\frac{n}{2^2}$ $\frac{n}{2^2}$ $\frac{n}{2^2}$
$2^2 n$
1
$\#\text{leaves} = 4^{\log_2 n} = n^2$
n^2
$\text{total} = O(n^2)$

问题3(d)–提示

Set $h = \log_2 n$

$$
\begin{aligned}
T(n) &= n + 4T(n/2) \\
&= n + 2n + 4^2 T(n/2^2) \\
&= n + 2n + 2^2 n + 4^3 T(n/2^3) \\
&\cdots \\
&= n + 2n + 2^2 n + \cdots + 2^{h-2} n + 2^{h-1} n + 4^h T(n/2^h) \\
&= n(1 + 2 + 2^2 + \cdots + 2^{h-1}) + 4^h T(n/2^h) \\
&= n\frac{2^h - 1}{2 - 1} + 4^h T(n/2^h) \\
T(n) &= n(n-1) + n^2 T(1) = O(n^2)
\end{aligned}
$$

问题3(e)-提示

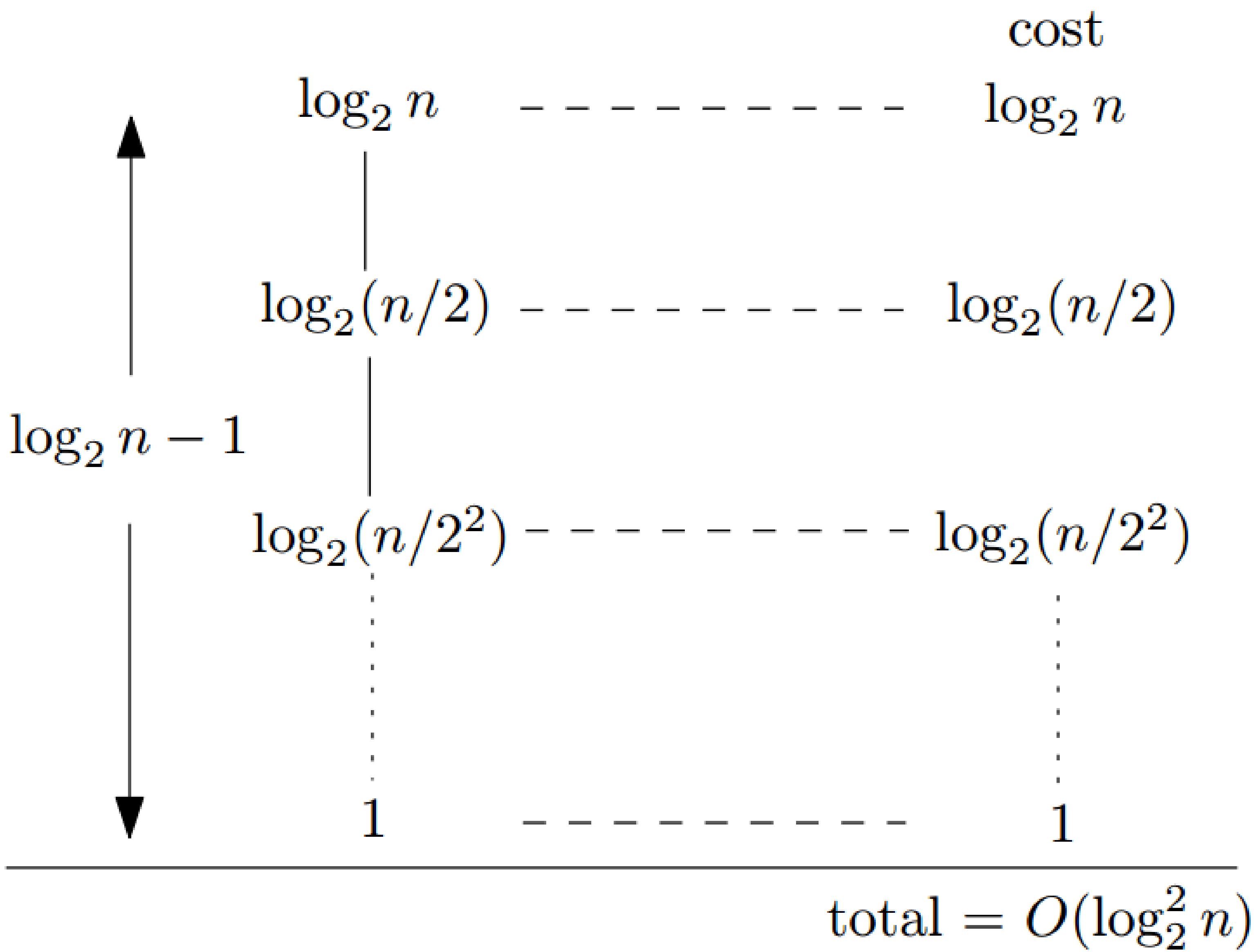

cost
n^2
n^2
$(\frac{n}{2})^2$ $(\frac{n}{2})^2$ $(\frac{n}{2})^2$
$3(n/2)^2$
$\log_2 n$
$(\frac{n}{2^2})^2(\frac{n}{2})^2(\frac{n}{2})^2$ $(\frac{n}{2^2})^2(\frac{n}{2})^2(\frac{n}{2})^2$ $(\frac{n}{2^2})^2(\frac{n}{2})^2(\frac{n}{2})^2$
$3^2(n/2^2)^2$
1
$\#\text{leaves} = 3^{\log_2 n} = n^{\log_2 3}$
$n^{\log_2 3}$
$\text{total} = O(n^2)$

问题3 (e) –提示

Set $h = \log_2 n$

$$
\begin{aligned}
T(n) &= n^2 + 3T(n/2) \\
&= n^2 + 3(n/2)^2 + 3^2 T(n/2^2) \\
&= n^2 + 3(n/2)^2 + 3^2(n/2^2)^2 + 3^3 T(n/2^3) \\
&\cdots \\
&= n^2 + 3(n/2)^2 + 3^2(n/2^2)^2 + \cdots + 3^{h-2}(n/2^{h-2})^2 \\
&\quad + 3^{h-1}(n/2^{h-1})^2 + 3^h T(n/2^h) \\
&= n^2[1 + 3/4 + (3/4)^2 + \cdots + (3/4)^{h-1}] + 3^h T(n/2^h) \\
&= n^2 \frac{1 - (3/4)^h}{1 - 3/4} + 3^h T(n/2^h) \\
&= 4n^2(1 - n^{\log_2 (3/4)}) + 3^h T(n/2^h) \\
&= 4n^2(1 - n^{(\log_2 3 - \log_2 4)}) + 3^h T(n/2^h) \\
&= 4n^2 - 4n^{\log_2 3} + 3^h T(n/2^h) \\
T(n) &= 4n^2 - 4n^{\log_2 3} + n^{\log_2 3} T(1) = O(n^2)
\end{aligned}
$$

问题3(f)-提示


cost
$\log_2 n$
$\log_2 n$
$\log_2(n/2)$
$\log_2(n/2)$
$\log_2 n - 1$
$\log_2(n/2^2)$
$\log_2(n/2^2)$
1
1
total $= O(\log_2^2 n)$

问题3(f)–提示

Set $h = \log_2 n - 1$

$$
\begin{aligned}
T(n) &= \log_2 n + T(n/2) \\
&= \log_2 n + \log_2(n/2) + T(n/2^2) \\
&= \log_2 n + \log_2(n/2) + \log_2(n/2^2) + T(n/2^3) \\
&\cdots \\
&= \log_2 n + \log_2(n/2) + \log_2(n/2^2) + \cdots + \log_2(n/2^{h-2}) \\
&\quad + \log_2(n/2^{h-1}) + T(n/2^h) \\
&= h \cdot \log_2 n - [\log_2(2) + \cdots + \log_2(2^{h-2}) + \log_2(2^{h-1})] \\
&\quad + T(n/2^h) \\
&= h \cdot \log_2 n - [1 + 2 + \cdots + (h-1)] + T(n/2^h) \\
&= h^2 + h - h \cdot (h-1)/2 + T(n/2^h) \\
&= h^2/2 + 3h/2 + T(n/2^h) \\
T(n) &= \frac{(\log_2 n - 1)^2}{2} + 3\frac{\log_2 n - 1}{2} + T(2) = O(\log_2^2 n)
\end{aligned}
$$